2016-ENST-0060

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité Communication et Électronique

présentée et soutenue publiquement par

Panagiotis MATZAKOS

le 24 octobre 2016

Techniques d' Ordonnancement et Gestion de la Congestion pour la Provision de QoS aux Réseaux DTN

Directeur de thèse: **Christian BONNET**

Jury

M. Tijani CHAHED, Professeur, TELECOM SudParis	Président du jury
M. Chadi BARAKAT, Chargé de recherche, INRIA Sophia-Antipolis	Rapporteur
M. Jörg OTT, Professeur, TUM University	Rapporteur
M. Vania CONAN, Directeur de recherche, Thales Communications	Examinateur
M. Thrasyvoulos SPYROPOULOS, Maître de conférences, EURECOM	Examinateur

TELECOM ParisTech

école de l'Institut Télécom - membre de ParisTech

T H È S E

DISSERTATION

In Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
from Telecom ParisTech

Specialization

Communication and Electronics

Ècole doctorale Informatique, Télécommunications et Èlectronique (Paris)

Presented by

Panagiotis MATZAKOS

Scheduling and Congestion Management Policies for QoS Provision in Disruption Tolerant Networks

Defense scheduled on October 24^{th} 2016

before a committee composed of:

Prof. Tijani Chahed	President of the Jury
Dr. Chadi Barakat	Reporter
Prof. Jörg Ott	Reporter
Dr. Vania Conan	Examiner
Prof. Thrasyvoulos Spyropoulos	Examiner
Prof. Christian Bonnet	Thesis Supervisor

THÈSE

présentée pour obtenir le grade de
Docteur de
Telecom ParisTech

Spécialité

Communication et Electronique

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Panagiotis MATZAKOS

Techniques d' Ordonnancement et Gestion de la Congestion pour la Provision de QoS aux Réseaux DTN

Soutenance de thèse prévue le 24 octobre 2016

devant le jury composé de:

Prof. Tijani CHAHED	Président du jury
Dr. Chadi BARAKAT	Rapporteur
Prof. Jörg OTT	Rapporteur
Dr. Vania CONAN	Examinateur
Prof. Thrasyvoulos SPYROPOULOS	Examinateur
Prof. Christian BONNET	Directeur de thèse

Abstract

Direct communication among wireless/mobile terminals (e.g., user devices, sensors) is being adopted more and more nowadays. It can be exploited in the context of local services provision, or as a means of complementing centralized communication (e.g., based on cellular networks) either in cases where the latter is absent/limited (e.g., after disaster scenarios, regions without coverage), or as a means of data traffic offloading. In this context, mobile terminals can be used as relays which forward other terminals data towards the destination (multi-hop paradigm). In many of the envisioned application frameworks, though, there exist communication challenges originating from the fact that end-to-end connectivity between the communicating peers cannot be maintained. This situation can be caused by reasons such as nodes mobility, poor channel conditions intermitting the wireless links frequently and/or for long periods of time etc. In such environments, dominant Internet approaches for reliable data transfer, based on end-to-end timely received feedback (e.g., TCP/IP related), can be inefficient.

Delay and Disruption Tolerant Networking has been aiming to tackle such communication challenges for a diverse set of mobile/wireless networking environments and applications. Among its basic concepts, DTNs support data storage at intermediate hosts, as a means of application sessions surviving connectivity disruptions. Based on this principle, data can be *stored* and *forwarded* throughout a sequence of multiple nodes, constituting the routing path to the data destination. In this context, *reliable data transfer* can be ensured between the consecutive storage points (i.e., on a *hop-by-hop basis*), comprising the basic alternative to end-to-end reliability provision in classic networks. This approach can be efficient for environments where the network topologies are predetermined and the timing of communication opportunities can also be known approximately (i.e., scheduled node contacts). The situation gets more complicated with opportunistic contact scenarios, though, where the nodes mobility patterns do not allow to know and exploit the timing of communication opportunities in advance. As a result, it is usually hard to maintain up-to-date routing information regarding how to reach the intended data recipients. On top of that, in cases with limited resources availability (e.g. local nodes storage, contacts duration) local buffer congestion events should be expected. In this context, ensuring data delivery is more challenging.

Moreover, in many envisioned DTN environments, it is expected that mobile nodes will be launching multiple applications in parallel. To this end, in resource constrained network settings, the generated data of some services may be more important to deliver than others. Furthermore, although the application framework that we focus on is generally tolerant to delays, it should be anticipated that data delivery of some applications is more urgent than others.

In this context, the primary focus of this thesis is on dealing with prioritization aspects among different application classes, with the aim of satisfying their individual QoS delivery requirements, in resource constrained DTN environments. These requirements are expressed either with respect to minimum required delivery ratio, or maximum accepted average delivery delay per QoS class. Towards this direction, we formulate the prioritization problems as distinct constrained optimization problems with respect to the delivery metric of interest (i.e., delivery rate maximization or delivery delay minimization delay). Then, we propose a distributed solution framework, which is based on optimizing the local scheduling and buffer management decisions during limited contact periods and buffer congestions, respectively.

More specifically, in chapter 1 we introduce the DTNs framework, highlighting the shortcomings of traditional TCP-like data transfer that it aims to solve, as well as the dependency of the suggested solutions space from the respective networking environments. In this context, we point out the motivation for our work. In chapter 2, we provide a classification of the existing congestion control and reliability assurance approaches for opportunistic contact networks in the literature. Through this classification, the aim is to highlight and provide a reasoning for the design choices with respect to our distributed optimization framework.

In chapter 3 we focus on two specific use case scenarios of interest which can benefit from our QoS prioritization policies. Both scenarios refer to opportunistic contact settings, with mobile nodes launching multiple applications in parallel, generating and propagating data of different QoS importance. The first scenario corresponds to an open field military mobility context and the second one to urban vehicular mobility. By indicating the differences of the two scenarios with respect to their corresponding services and mobility considerations, we aim to highlight the suitability of our policies which can be applied in a wide range of opportunistic use cases.

In chapter 4 we analytically present our distributed scheduling and buffer management schemes, corresponding to the two aforementioned constrained optimization problems. In this context, we claim that the optimal intended behavior should be the following. Depending on the resources availability, we should first aim for application classes delivery requirements satisfaction, in the order of their importance, and, then, for the overall network performance maximization, with respect to the metric of interest. In practice, the latter performance target is equivalent to preventing the "starvation" of lower application classes. To this end, our schemes are based on extracting appropriate per message utilities, derived from the per-message delivery prediction expressions, which are used to locally determine their optimal scheduling and dropping sequence during node contacts and buffer congestion events, respectively.

The delivery predictions in chapter 4 are based on a mobility model which considers *exponentially distributed* inter-meeting times and *homogeneous* contact patterns between the distinct pairs of nodes. Homogeneity in this context refers to approximating all the meeting rates of node pairs with a common rate $\tilde{\lambda}$ which is a characteristic of each mobility trace. Although this assumption is valid for simple mobility models (e.g, random waypoint), it can be inaccurate for real life mobility, thus affecting the precision of the delivery predictions and consequently the delivery performance of our prioritization policies. To this end, in chapter 5, we make the necessary adaptations in our exponential based models of inter-contact times, in order to account for *heterogeneous* and *sparse contact networks* (i.e., scenarios where there is the possibility that some nodes will never encounter each other). The performance benefits of our extensions, are demonstrated through simulation results based on different real mobility traces. In the same context of being aligned with real life mobility and node contact patterns, we have also considered to remove the exponentiality assumption and model the pairwise inter-meeting times through the generalized pareto distribution (power-law family) in appendix A. The performance with this approach is evaluated through comparisons with the exponential based approaches.

In chapter 6 we conclude the thesis and discuss about future research directions.

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Christian Bonnet for giving me the opportunity to work on this topic and collaborate with esteemed industrial partners, in the context of a European project. His guidance and support, both in academic but also in personal level, meant a great deal to me during the whole period of my thesis.

Also, I would like to express my gratitude to Prof. Thrasyvoulos Spyropoulos. His guidance in defining and dealing with challenging research problems was invaluable for me, both for completing the work of this thesis, but also from a training perspective.

Moreover, I would like to express my strong appreciation to Prof. Jérôme Härri for guiding me through my very first research steps during my Master thesis and the beginning of my PhD. I will always remember that my work with him was the one which led to the start of my thesis.

A PhD is usually considered a long and lonely road and to some extent this might be true. But for me, it was also a time when I had the chance to make friendships that will last for life, sharing moments that I will always remember. First, I would like to thank my family in Nice, Dorine, Nikos and Katerina, who have always been there for me and with whom I spent so many laughing moments and had the most amazing trips. Also many thanks go to all the friends I made here, for the great times we had together: my crazy officemate Giorgos, Miltos, Pavlos, Laura, Remi, Akis, Dimitris, Dora, Loukas, Katt, Giorgos, Robin and many others; but also to my old friends who have always been supportive despite the distance: Alexandra, Babis, Dio, Giannis, Manos, Myrto and Paris.

I will always be extremely grateful to my parents Kostas and Roula and my brother Giannis for loving and supporting me in every way they could through my whole life.

Last but not least I want to dedicate this thesis to my beloved Dorine for standing next to me. Through her deepest love, support and understanding she could always make the difficult times, easier and give me the strength to carry on.

Contents

List of Figures

List of Tables

Acronyms

Here is the list of acronyms used in the text.

4G	Fourth generation.
ACK	Acknowledgment.
ADD	Average Delivery Delay.
BDR	Bundle Delivery Ratio.
BFT	Blue Force Tracking.
BPA	Bundle Protocol Agent.
BRA	Bundle Routing Agent.
BSS	Basic Service Set.
C2C	Car to Car.
C2V	Cloud to Vehicle.
CCDF	Complementary CDF.
CCTV	Closed Circuit Television.
CDF	Cumulative Distribution Function.
CL-ACK	Congestion Level based ACK.
CQI	Channel Quality Indicator.
DCC	Decentralized Congestion Control.
DMM	Distributed Mobility Management.
DSRC	Dedicated Short Range Communication.
DS-TP	Deep Space Transport Protocol.
DTN	Delay/Disruption Tolerant Networks.
DTNRG	Delay Tolerant Networking Research Group.

EDA European Defence Agency.

EMCON Emission Control.

ETSI European Telecommunications Standards Institute.

FCD Floating Car Data.

GAS Geographic Address Set.

GBSD Global Knowledge Based Scheduling and Drop.

GMM Geographic Mobility Management.

G-SACK Global SACK.

HBSD History Based Scheduling and Drop.

HoA Home Address.

ICN Intermittently Connected Network.

IF-Node Infrastructure Node.

IP Internet Protocol.

IPv6 Internet Protocol version 6.

ITS Intelligent Transportation Systems.

IF-Node Infrastructure Node.

KKT Karush Kuhn Tucker

LIMME Location and Infrastructure Mobility Management Entity.

LM Location Management.

LTE Long Term Evolution.

LTP Licklider Transmission Protocol.

LTP-T LTP Transport.

LU Location Update.

MANET Mobile Ad-Hoc Network.

MIDNET Military Disruption Tolerant Networks.

MIPv6 Mobile IPv6.

MN Mobile Node.

MP-TCP Multi-Path TCP.

MTC Machine Type Communication.

NACK Negative Acknowledgment.

NORM NACK Oriented Reliable Multicast.

ORWAR Opportunistic Routing with Window Aware Adaptive Replication

PDR Packet Delivery Ratio

POI Point of Interest.

PRoPHET Probabilistic Routing Protocol using History of Encounters and Transitivity.

QoS Quality of Service.

RLC Random Linear Combination.

RSU Roadside Unit.

SA Service Advertisement.

SACK Selective ACK.

SI Service Interest.

SLAW Self-similar Least Action Walk

SnW Spray and Wait.

SUMO Simulation of Urban Mobility.

TBCC Token Based Congestion Control.

TCP Transmission Control Protocol.

TCPCL TCP Convergence Layer

TTL Time to Live

UDP User Datagram Protocol.

UMTS Universal Mobile Telecommunications System.

V2I Vehicle to Infrastructure.

V2V Vehicle to Vehicle.

WAVE Wireless Access in Vehicular Environments.

WiFi Wireless Fidelity.

Chapter 1

Introduction

1.1 Delay and Disruption Tolerant Networks scope

Mobile communication capabilities nowadays keep increasing in terms of connectivity and data rates provision, as well as in terms of the wide range of applications they provide. Users can exploit different types of interfaces (e.g. 4G/LTE, WiFi etc.) and network types (e.g. infrastructure-based, infrastructure-less), depending on the application type (e.g. internet based or local scope) and the networking conditions (e.g. increased network traffic through one interface can trigger user traffic offloading through another interface).

Mobile Ad-hoc networks (MANETs) are a family of networks characterized by direct communication among moving users without any support from centralized infrastructure (e.g. cellular base stations or wireless access points). Thus, any node (user) can act as data source, destination, or a relay assisting to convey a message to its destination. In this context, the use of MANETs is popular for services of local scope (e.g. vehicular ad-hoc networks for traffic/safety applications, wireless sensor networks, military/tactical networks, mobile social networks etc.). Due to the movement of the users, the topology of such networks changes dynamically and thus building data routing paths between source and destination pairs of nodes is challenging. However, it is assumed that there always exist end-to-end routing paths from any source to any destination node. Nonetheless, there are real world scenarios for which such an assumption would be unrealistic, due to the frequent and/or long connectivity disruptions among the communicating peers, leading to intermittently connected networks (ICNs). Such disruptions may result from sparse network topologies, terrain obstacles, nodes mobility or resource constraints (bandwidth per communication opportunity, storage, energy).

To this end, **Disruption Tolerant Networks (DTNs)** can be considered as a special type of MANETs which aims to provide communication services under such stressing conditions. Delay/Disruption Tolerant Networking (DTN) [1] was initially dealing with communication challenges in Space (Interplanetary Deep Space, Satellite communications etc.). Such challenges usually refer to maintaining end-to-end data delivery alive when experiencing large propagation delays, or delays caused by periodic loss of line of sight conditions between the communicating peers. The application domain of DTNs has more recently been enriched with a large family of terrestrial networking environments, under networking conditions which have the same impact with the ones encountered in space communications: the loss of continuous end-to-end connections.

In order to survive intermittent connectivity, DTNs are based on the store-carry-and forward concept: the mobile nodes can store their own, or other nodes contents until some next communication opportunity appears, either with the destination (content delivery) or with some relay node to which they can convey data. Based on this principle, the DTN applications running on the end-hosts can remain transparent of the connectivity disruptions. However, the lack of continuous end-to-end connectivity, the limited communication opportunities, as well as the requirement for nodes to store their own and other nodes data in resource-constrained environments, makes it very hard to guarantee data delivery within a specific time limit.

In this context, data transport based on TCP/IP proves to be highly inefficient in most cases. Transmission Control Protocol (TCP) [2] and multiple of its extensions (e.g., [3], [4], [5]) are very popular for many Internet-oriented applications that require reliability (i.e. ensuring data delivery to the intended recipient(s)) and provision for congestion control. However, its operation is based on the existence of continuous end-to-end connectivity, which is usually absent in the aforementioned types of network environments. Moreover, on the one hand, TCP mechanisms cannot tolerate the large delays induced by intermittently connected networks. On the other hand, the "chatty" TCP operations (e.g. connection establishment, feedback based congestion control) usually consist a very large overhead for DTNs which can degrade their performance significantly, considering the limited amount and duration of communication opportunities.

Depending on the network environment, delay tolerant networking can give functional alternatives to TCP-like approaches, in order to provide reliable communication, when this is feasible, or best effort approaches when this is not feasible. Such best effort approaches might not be able to guarantee data delivery, however their goal is to maximize the network performance, given the aforementioned stressing conditions.

In the remainder of this chapter, we first discuss on the special characteristics of different DTN settings. Then, we provide some more details on the traditional data transport requirements and challenges comparing to TCP/IP world, which motivated our work and formulated the solution space for the problems that we investigate. Finally, we summarize the main contributions of this thesis.

1.1.1 Networking environments and mobility patterns

Mobile nodes can exchange data when they are within communication range with each other (i.e. when they come in *contact*, based on DTN terminology) and the external conditions permit to do so. In this context, different types of contacts can take place, based on the mobility patterns of the nodes:

- *Scheduled/Predetermined contacts:* the pairs of nodes come within range distance, based on a specific schedule which can be known. Such type of contacts are usually observed in inter-planetary networks, where the communicating parts (e.g. satellites) move based on predetermined orbits and, as a result, the contact opportunities appear periodically. Use case scenarios where the contacts are predetermined can be found in terrestrial network settings as well. For example, in [6], [7] "message-ferries" are used to receive and convey users data to gateways, thus providing cheap Internet access. In such scenarios, data routing can be static and even end-to-end paths between the source and the destination can be exploited. Thus, the main stressing factor here originates from the large delays caused by the duration between consecutive contacts (*inter-contact times*), or the propagation time and channel errors in inter-planetary networks.

- *Opportunistic contacts:* the pairs of nodes move without any specific schedule and/or route, so they come in contact and can exchange data during unexpected opportunities. In this context, it is impossible to know when the next contact with a specific node will take place. As a result, it is also usually impossible to build valid and up-to-date routing paths to destination nodes. Opportunistic type of contacts can be found in a wide range of terrestrial scenarios where DTN solutions are applied (e.g. mobile social networks, vehicular or military DTNs).

- *Probabilistic contacts:* We can consider probabilistic as a contact type which "bridges the gap" between the predetermined and the opportunistic type. Particularly the movement and the upcoming contacts of the mobile nodes are not known deterministically but at the same time they are not totally random. For example, in real life mobility, we can observe common tendencies among different individuals (e.g. people going to work with the same way every day (bus, train, car) following the same route and meeting some same people). Such tendencies can be gathered in the form of statistical history information of contacts. Then, they can be exploited to predict the probability of node meetings and use these predictions to construct probable data routing paths leading to the intended recipients.

Another stressing factor related to mobility is the duration of node contacts. Depending on how fast the nodes move, the available time (contact window) during which they can exchange data may differ significantly. As a result, we should expect that vehicle contacts in a highway for example would usually last less than pedestrian contacts. The length of the contact window, in coordination with the available bandwidth per contact and the utilized radio transmission range determine which amount of data can be exchanged. Thus, it is very likely that the data which require to be forwarded are more than the data that can actually be forwarded.

1.1.2 Data transport requirements and challenges comparing to TCP/IP

Depending on the application type and the required Quality of Service (QoS), two basic transport layer responsibilities are the following:

- *Reliability:* As already mentioned, reliability refers to the capability of ensuring complete end-to-end data delivery. In TCP this is done based on timely received end-to-end control feedback, in the form of data segment acknowledgments (ACKs) which are forwarded on the reverse path (i.e. from the recipient of successfully received data segments to the initial source). In ICNs such a practice for ensuring reliability would suffer from very large delays and probably fail, due to the lack of stable end-to-end paths, especially if we consider opportunistic type of contacts.

- *Congestion and flow control:* In TCP terminology, these two mechanisms should manage the amount of data traffic load, to avoid overloading the network and the specific receiver of a data stream, respectively [2], [8], [9]. Thus, congestion control would be responsible for not overburdening a router which could be the "bottleneck" for many connections running concurrently. Flow control, on the other hand, should be triggered from the receiver side of a connection to make sure that the sender does not transmit more traffic than the one the receiver's buffer can handle locally. In that sense, those two mechanisms assist in ensuring reliability. However, as for the case of reliability, they are based on the timely reception of end-to-end control feedback. Thus, delayed or not received ACKs are considered as congestion indicators, which trigger the decrease of the transmission rate at the sender side, as a way to handle congestions [8]. In a DTN environment, however, there is high chance that the reason for the end-to-end delays is the intermittent connectivity, rather than the network congestion. Differentiating the losses due to intermittent connectivity from the ones due to congestion is a challenging task, which popular TCP-based approaches are usually not designed to consider. As a result, their use in such environments may lead to not exploiting the already limited available resources adequately.

1.1.3 DTN solutions framework and relation with data routing

As highlighted earlier, the DTN community has promoted the store-carry-and forward approach, based on which, data messages are conveyed through multiple relay nodes before reaching the final recipient(s). In this context, hop-by-hop reliability is suggested, as a feasible alternative to end-to-end reliability. Specifically, a message is acknowledged at a previous hop (relay) when moving to the next. Then, the new relay stores the message and it is responsible either for its delivery to the destination, or for its forwarding to the next hop of the routing path. Based on this framework, an end-to-end delivery path can be split in multiple sub-paths, where, for each sub-path, reliability, from the perspective of ensuring the delivery of data can be ensured. Also, if the timings of communication opportunities among the DTN nodes comprising the end-to-end path are more or less known, the data delivery delay to the destination can also be estimated. Thus, this approach can be adequate for mobility scenarios where the the end-to-end routing paths may not be continuous, but they are either static (scheduled contacts), or they can be discovered with some high probability (probabilistic contacts). However, as the randomness in node contacts is increasing (opportunistic contacts), the difficulty in discovering valid routes to content destinations makes it more complicated to guarantee data delivery, let alone within specific time limits.

In order to increase the probability of content delivery in opportunistic networks, a usual practice in DTN literature is to use multiple-copy, instead of single copy routing schemes (e.g. [10], [11]). In this context, various routing paths can be created randomly, with the purpose of one of the copies meeting the destination. It is evident that this approach can also decrease the delivery delay: i.e., elapsed time until the delivery of the first message copy. However, keeping track of multiple paths and/or sub-paths, in order to ensure reliability in the manner described before, would be more complicated and require more control information overhead now. Moreover, if we consider resource constrained environments (e.g. Energy/Buffer limited wireless sensor networks [12], vehicular networks with limited contact durations and/or buffer limited [13], military DTNs [14]), disseminating multiple replicas per message in a DTN network can increase dramatically the overall traffic load comparing to the available resources.

1.2 Motivation

According to the previous discussion, delivery reliability can refer to the following application requirements: ensuring that the disseminated data will be eventually delivered to the recipient and making sure that data delivery will take place within some more or less tight time limit. In this context, there is a clear differentiation regarding the challenges encountered in different networking environments, with respect to the node contact patterns. Starting from the unsuitability of TCP based solutions, for both types of environments, due to their aim for reserving resources for the whole end-to-end path, we argued why hop-by-hop reliability can be functional for scheduled contact environments. However, we also pointed out that the efficiency of this approach is questionable for opportunistic contact environments.

For such environments, a common practice in the literature is to combine multiple copy routing with local scheduling and buffer management algorithms. The aim is to optimally decide on how much and which data to replicate, during limited contact opportunities, and how to deal with buffer congestions, respectively (e.g., [15], [16]). Although not ensuring reliability according to the aforementioned definition, such approaches intend to maximize the performance (e.g. minimize delivery delay, maximize delivery ratio), given a set of resource constraints.

Nevertheless, such schemes usually assume resources contention among data sessions of equal importance. In many envisioned DTN scenarios, however, mobile nodes are expected to launch multiple applications in parallel. In this context, ensuring successful data delivery, or minimizing the delivery delay may be more important for one DTN application than for another. Thus, multiple classes of QoS requirements can be defined and exist in parallel. Making sure that the individual classes are satisfied when the available resources permit so can be considered as a "loose" equivalent of QoS reliability provision in end-to-end continuously connected networks, where the respective requirements should always be captured.

However, prioritization among satisfying different QoS requirements, in resource constrained DTNs, is an open issue. A simple scheme, for example, could give absolute priority to applications of higher classes. However, if prioritization is based purely on the QoS class, then applications belonging to lower classes would "starve" (i.e. they would always be the last to be scheduled and the first to get dropped).

1.2.1 Use case scenarios

As highlighted previously, efficient QoS provision for multiple concurrently launched application classes can be a significant requirement in different use case scenarios. It should be expected, though, that such use cases can refer to application scenarios which are completely disconnected with each other. Accordingly, the mobility patterns can also differ significantly. In this context, it is important to come up with reliable QoS provision policies, which can be plugged in diverse DTN use cases and fulfill the respective requirements. To this end, we highlight here two such distinct use cases which could benefit from such generic policies.

1.2.1.1 Military tactical Networks

As stated earlier, QoS-based data prioritization can be applied in different use case scenarios. In the context of European Defence Agency (EDA) Military Disruption Tolerant Networks (MIDNET) project, our focus was on military tactical scenarios. Military networks can leverage the functionality of MANETs, as they can bring higher data rates and ease of operation in the front-line of the battlefield (e.g. configuration, management). Thus, they are envisioned as a means of providing inter-connection capabilities among different coalition forces (e.g. ad-hoc networks belonging to different countries), or with other existing types of networks (thus extending connectivity). However, the capabilities offered with MANETs should not hide the challenges due to connectivity disruptions which are present in battlefield environments. Apart from the *physical* causes of disruptions, originating from the nature of the network as specified earlier, military

MANETs have to cope with *artificial* disruptions, as well, dictated by operational purposes (e.g. radio jamming which hinders proper communication, imposition of emission control (EMCON) for some periods of time, forbidding any transmissions and information exchange) and leading to large network disconnection periods.

Targeting such networking environments, the purpose of MIDNET was to propose solutions based on the DTN paradigm, in order to maintain data sessions alive during disconnections, while ensuring a smooth and transparent transition to Internet Protocol (IP) connectivity, when the network conditions allow to do so. In this context, tactical/operational requirements were first mapped to networking functional requirements. Among these requirements, there is the need for being able to prioritize some data traffic over some other, under the aforementioned resource constraints (i.e., storage, energy, amount and duration of communication opportunities). The competition may refer to traffic of the same type (e.g. position updates obsoleting older positioning information), or more complicated scenarios where the traffic originates from different applications which are being launched concurrently at the DTN nodes.

Indeed, in tactical networks multiple classifications for different types of services can be made, with respect to:

- Their nature (e.g. voice communications, video streaming, messaging).
- Their objective (e.g. command data, shared situational awareness, status reports [17]).
- Their criticality (i.e., real time, non-real time but time critical, non-real time-lower priority, best effort, as defined in [18]).

Defining policies to determine the prioritization order among such different types of services is not straightforward and it was one of the challenges that MIDNET had to address, in the context of QoS driven routing.

1.2.1.2 Floating Car Data upload in Hybrid Vehicular Networks

In the context of an external contract with Orange Labs, we studied the applicability of DTN solutions in hybrid vehicular networks and proposed a framework for the use case of uploading Floating Car Data (FCD) through Infrastructure nodes, residing at the edge of the network (i.e., Dedicated Short Range Communications (DSRC) Roadside Units (RSUs) or Cellular Base Stations). FCD applications generally refer to the collection of large amounts of highly dynamic data, originating from vehicles. Such data can refer to localization information (position, speed, movement direction) that are useful for traffic management [19], or sensor data, useful for maintenance operations and statistics collection for the car manufactures [20]. Thus, FCD applications can leverage from the DTN framework in the context of surviving disruptions due to any of the following events:

- Temporal loss of connectivity while within the coverage of the same Infrastructure node (short disruption).
- Longer loss of connectivity while moving from the coverage of one Infrastructure node to the next available one (long disruption).
- The local Infrastructure is overloaded and cannot accept any more FCD. In this case an “artificial” disruption is caused.

Considering the heavy loads of data traffic envisioned to originate from different application types at the vehicles (e.g. traffic management, safety), FCD applications are supposed to produce non-critical type of traffic. However, resources permitting, appropriate prioritization policies are required to prevent their starvation, either in cases of competition with other application types, or in cases of internal competition among different FCD applications.

1.2.2 Real world mobility challenges

In the context of multiple copy routing and opportunistic DTNs, there exist scheduling and buffer management approaches which are based on their per message predictions for a metric of interest (e.g. expected delivery probability/delay), to optimize data prioritization decisions (e.g., [15], [16]). Thus, a simple policy could favor messages whose estimated delivery probability is lower than others, for example. To make accurate predictions, though, it is important to capture the contact patterns among distinct node pairs adequately. To clarify this, consider that the delivery delay of a message can be expressed as *the time needed for the next contact of one of the copy holder nodes, with the destination.*

One popular approach, in opportunistic DTN routing protocols, is to consider that the inter-meeting times characterizing each pair of nodes $< i, j >$ are independent from each other and can be modeled through the exponential distribution with a common rate parameter $\tilde{\lambda}$, approximating all individual rates $\lambda_{i,j}$ (assuming *homogeneous contact networks*). The exponentiality argument is supported by existing works, showing that a lot of mobility models and real traces correspond to contact models with approximately exponential tails [21], [22], [23]. However, the existence of strong power-law components in the inter-meeting time distributions of popular real traces is also shown. Moreover, in real mobility, a large variance on the meeting rates between different pairs of nodes can be expected (*heterogeneous contact networks*) [24], [25], [26]. Finally, some pairs of nodes might never encounter each other during the duration of a mobility trace, leading to *sparsely connected contact graphs.* In such scenarios, the aforementioned exponential and homogeneous mobility model may lead to inaccurate estimations of inter-meeting times.

1.3 Contributions and Outline

The main focus of this thesis is on providing QoS performance guarantees for different data traffic classes in the DTN context, when the available resources permit to do so. The QoS requirements are mapped either to delivery ratio or delivery delay. In that sense, as mentioned earlier, the solution framework that we apply can be considered as a means of providing a loose equivalent of QoS provision and reliability in end-to-end connected networks. We focus on opportunistic contact settings, which, IPNs excluded, characterize the majority of real life mobility scenarios where the DTN functionality is useful, and, at the same time, the most challenging ones. In this context, our QoS provision framework is based on open-loop (i.e., without using any kind of data acknowledging mechanism) distributed scheduling and buffer congestion management, using local decisions based on delivery predictions, to optimize the overall network performance. Particularly, we consider that an optimal distribution of (the limited) resources would: *(i) make sure that the individual QoS requirements are satisfied, when this is feasible based on the resources availability* and *(ii) allocate any remaining resources optimally, to maximize the desired performance metric.*

In the following, a discussion regarding the simulation tools we used to evaluate our work, as well as the outline of the dissertation chapters and the corresponding main contributions is provided.

Simulation tools

At the beginning of this thesis, we had been working on extending a version of NS-3 [27], which is provided as the Network Simulator module of the iTETRIS simulation platform [28]. iTETRIS is a powerful tool, allowing to launch realistic vehicular communication and mobility simulations, according to the defined scenarios. Although considering to use it for our DTN related evaluations, the scope of the problems we finally decided to focus on rendered its use rather impractical. Based on this scope, the detailed simulation of communication protocol operations and the precise reproduction of mobility trajectories, provided by the complex simulation framework of iTETRIS, were not necessary for our purposes. However, these capabilities of iTETRIS were exploited in the context of a different class of problems that we dealt with, targeting at geo-assisted IPv6 mobility management for the support of vehicular Cloud services. Our contributions with respect to this topic, together with the related simulator extensions we provided, are presented analytically in appendix B, as an independent work from the rest of this thesis.

Going back to the discussion regarding our DTN evaluation framework, we have built a simulation environment based on Matlab. This environment uses nodes contact traces (i.e., list of node contacts) as inputs, together with the respective pairwise contact statistics, which can be extracted from synthetic (i.e., according to artificial mobility models) or real life mobility traces. The node contacts trigger data exchanges between the mobile nodes, which prompt the scheduling and buffer management operations related to our policies. To this end, the amount of data which can be exchanged during node contacts can be restricted according to the applied configuration, with the aim of considering the limited duration of node meetings.

Chapter 2 - Related Work

In this chapter we first present the generic architecture that has been proposed by the Delay Tolerant Network Research Group (DTNRG) community to support the basic principles of delay tolerant networking. Then, we focus on the congestion control and reliability aspects in DTNs, first showing the dependency of each solution framework from the mobility environment and more specifically on the node contact patterns where it is applied, as specified earlier. We review and classify existing schemes in the literature based on multiple criteria. Through this process, we end up justifying the general framework we used, to deal with the QoS prioritization problems, for the network scenarios of interest.

The work related to this chapter appears in:

- *P. Matzakos, C. Bonnet, "A taxonomy of congestion control and reliability approaches in opportunistic DTNs", Research Report RR-16-323, Eurecom, September 2016.*

Chapter 3 - Use case scenarios

The use case scenarios highlighted in section 1.2.1, are reviewed here in detail. Particularly, we focus on their distinct application contexts and corresponding mobility considerations, indicating how the QoS prioritization framework that we indicate in chapter 2 can be exploited. In this context, we also provide a set of DTN architectural extensions on top of existing architectures for each use case, to support our framework from a practical implementation perspective.

Chapter 4 - Joint distributed scheduling and buffer management policies for DTN Applications of different traffic classes

We first formulate the problems of maximizing the network performance, with respect to each metric of interest (i.e. maximize average delivery rate, or minimize average delivery delay), as constrained optimization problems. The constraints correspond to distinct QoS class requirements and resource limitations. A centralized solution to this problem would be based on per message delivery estimations to perform a buffer space allocation, among the copies of non-delivered messages, that is feasible, i.e., ensures that at least the delivery requirement of each message is captured, and optimal, i.e., leads to performance maximization with respect to each metric. Particularly, it would derive an optimal allocation vector $\mathbf{n}^*$, whose entries would correspond to the number of copies for each individual message.

However, based on the nature of DTNs, such an approach is impossible because it implies the need for a centralized entity that would know and control the status of all existing messages, instantaneously. To this end, we focus on a distributed solution. The basic assumptions that are made and justified are the following: (i) the mean pairwise meeting rate of a given mobility scenario is considered known, (ii) the knowledge/estimation of dynamic message related parameters (e.g. per message number of copies) is also available. Thus, we derive optimal per message utilities, extending the ones derived in [16] for the QoS unconstrained problem, by adding appropriate penalty functions to account for QoS constraints violation. Based on the framework of our distributed algorithm, epidemic routing is used [10] and, at each meeting between two nodes, only a limited number of variables from the copies vector $\mathbf{n}$ can be affected (i.e., the ones corresponding to the messages residing in either of the two node buffers). Nevertheless, prioritizing messages based on their optimal utility ranking leads to the maximum performance gain among all feasible directions at each decision step, corresponding to a distributed implementation of a gradient ascent (or descent, depending on the objective) algorithm that eventually converges to the optimal solution. We validate the optimality of our approach through extensive simulations and comparisons with other prioritization policies, for homogeneous contact networks.

The work in this chapter corresponds to the following publication:

- *P. Matzakos, T. Spyropoulos and C. Bonnet, "Buffer Management Policies for DTN Applications with Different QoS Requirements," 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015.*

Chapter 5 - QoS prioritization improvements for heterogeneous contact networks

In chapter 4, we provide an optimal distributed solution to the problem of QoS prioritization in resource limited DTNs. A prerequisite for the fine operation of our policy is to make accurate predictions regarding the delivery performance, on a per message basis. In chapter 4, these predictions are based on the assumption of exponentially distributed inter-meeting times and homogeneous contact networks. In this chapter, we keep the exponentiality assumption, but we make the necessary extensions in our prioritization framework in order to account for heterogeneous and sparse contact networks, which better correspond to real life mobility. We show the performance benefits of these extensions through simulations with real mobility traces, by comparing them with the implementation of our scheme based on the homogeneity assumption. We also compare the performance of our policy with other policies, based on real traces.

Furthermore, we suggest an alternative approach based on using Spray-and-Wait [11], instead of Epidemic routing. This is a "one-shot" approach which, instead of dynamically monitoring the per-message delivery performance estimates, selects the maximum number of copies per message in an "optimized" way at the beginning of their lifetime. Although performing worse than our basic policy, as we show through simulation results, we claim that this approach can be attractive due to its ease of use since it doesn't require the availability of message related information during runtime.

The work in this chapter corresponds to the following submission:

- *P. Matzakos, T. Spyropoulos, and C. Bonnet, "Joint Scheduling and Buffer Management Policies for DTN Applications of Different Traffic Classes", submitted to IEEE Transactions on Mobile Computing, July 2016.*

Appendix A- Analysis of real mobility traces based on Pareto contacts and impact on QoS prioritization performance

In chapters 4 and 5 we were based on the exponentiality assumption regarding the pairwise inter-meeting times distributions for homogeneous and heterogeneous contact networks, respectively. However, as stated earlier, existing literature highlights the existence of strong power law components in these distributions for real mobility traces. In this context, we remove the exponentiality assumption here and evaluate the performance of our policy when the generalized Pareto model (i.e., power law family) is used instead.

To this end, we first describe the framework we used to extract the appropriate type of pairwise distribution for our purpose. Then, we compare the obtained prediction results when modeling this distribution through the exponential and the generalized Pareto models in scenarios without resource constraints, for different real mobility traces. For the majority of traces it seems that the Pareto based modeling outperforms the homogeneous exponential one (i.e., based on the formulations of chapter 4). When we went back to our resource constrained prioritization problem, though, and compared it also against the exponential based extensions of our algorithm accounting for heterogeneous contacts (i.e., based on the formulations of chapter 5), we noticed that it performed worse than the latter in the context of the delivery delay optimization problem and did not offer any performance benefits with respect to the delivery ratio optimization problem.

Appendix B- An IPv6 Architecture for Cloud-to-Vehicle Smart Mobility Services over Heterogeneous Vehicular Networks

As described earlier, the work described here is not related to the rest of the contributions of this thesis. Particularly, we provide the specification of a cloud-initiated Point-of-Interest (PoI) application, and illustrate its requirements for a convergence between IPv6 mobility management and Dedicated Short Range Communications (DSRC) geographic services. We propose to extend a flat IPv6 mobility management architecture with a new functional block, namely LIMME (Location & Infrastructure Mobility Management Entity), composed of three key functions: a Location Manager (LM) acting as location anchor point for cloud-based services, a Geographic Mobility Management (GMM) function acting as location proxy for the LM and handling IPv6 mobility, and an Infrastructure Node selector, which selects a route based on geographical data and local infrastructure node conditions. As a proof-of-concept, we implemented these extensions on the iTETRIS ITS simulation platform and illustrated their benefits in enhanced IPv6 mobility management and traffic offloading.

The work of this chapter is published in:

- *P. Matzakos, J. Härri, B. Villeforceix and C. Bonnet, "An IPv6 architecture for cloud-to-vehicle smart mobility services over heterogeneous vehicular networks," 2014 International Conference on Connected Vehicles and Expo (ICCVE), Vienna, 2014.*

Chapter 2

Related Work

2.1 Introduction

As discussed in chapter 1, providing guarantees for reliable data delivery is challenging for DTNs. In the current chapter, we first describe the general DTN Architecture framework (section 2.2), focusing on the basic functions of the introduced *bundle* and *convergence layers*, which aim to survive intermittent connectivity and provide interconnection within heterogeneous sub-networks. These functions include the hop-by-hop custody transfer, as a means of ensuring data delivery, in the absence of end-to-end connectivity. Although this approach is functional for scenarios with predetermined topologies and scheduled type of contacts between the DTN nodes, it might not be adequate for challenging scenarios characterized by more random mobility patterns and dynamically changing topologies, as highlighted in chapter 1. Moreover, due to the intermittent connectivity conditions, the limited duration of communication opportunities and the need for intermediate data storage for extended and possibly unpredictable amounts of time, data scheduling and storage congestion control are significantly important to coordinate the distribution of limited resources in the network.

In the framework of distinct applications QoS provision, we claim that both congestion control and reliability operations can be crucial. In this context, in section 2.3 we review and classify existing schemes in the literature for both types of operations. This classification is based on the associated networking environments, their objectives and their basic operation principles. The aim of this review is to "pave the way" towards the design choices we made with respect to our QoS prioritization policies which are highlighted in section 2.4 and will be analytically presented in the next chapters.

2.2 DTN Architecture

The DTN architecture [29] has initially been proposed to tackle the communication challenges appearing in interplanetary, deep space networks. However, the suggested framework was envisioned to consist the basis, on top of which, functional solutions for other types of networks (e.g., wireless terrestrial sensor networks, underwater, satellite) can be built, as well. Such networks may also suffer from intermittent connectivity, leading to frequent network partitioning and, eventually, the incapability of maintaining end-to-end connections active.

To overcome such obstacles, the DTN-architecture relies on a store-carry-and-forward, hop-by-hop or subnet-by-subnet data delivery strategy, depending on the length of the path which has to be traversed, before the transmitted data has to be locally stored to survive some sort of disruption. In Fig. 2.1, a simple scenario example leveraging from the DTN architecture is depicted. Based on this scenario, two of the intermediate nodes participating at the end-to-end data delivery path: i.e., custodian and moving data "ferry" nodes, have to store the data originating from a source node, until the next hop is discovered within communication range and the respective partitions are connected to the rest of the network. In this way, the end-to-end path to the destination can be split into multiple sub-paths. Apart from local connectivity disruptions, the need for intermediate storage and interconnection can be dictated by the presence of Heterogeneous sub-networks within the same network. In this context, heterogeneity can refer to different Network types (e.g. IP vs non-IP based subnets connected through gateway nodes, as shown in Fig. 2.1), or different locally experienced communication conditions (e.g. higher vs lower bandwidth radio interfaces). In such cases, compatibility with each Network specific stack (Fig. 2.2) is a prerequisite to provide seamless communication capability.

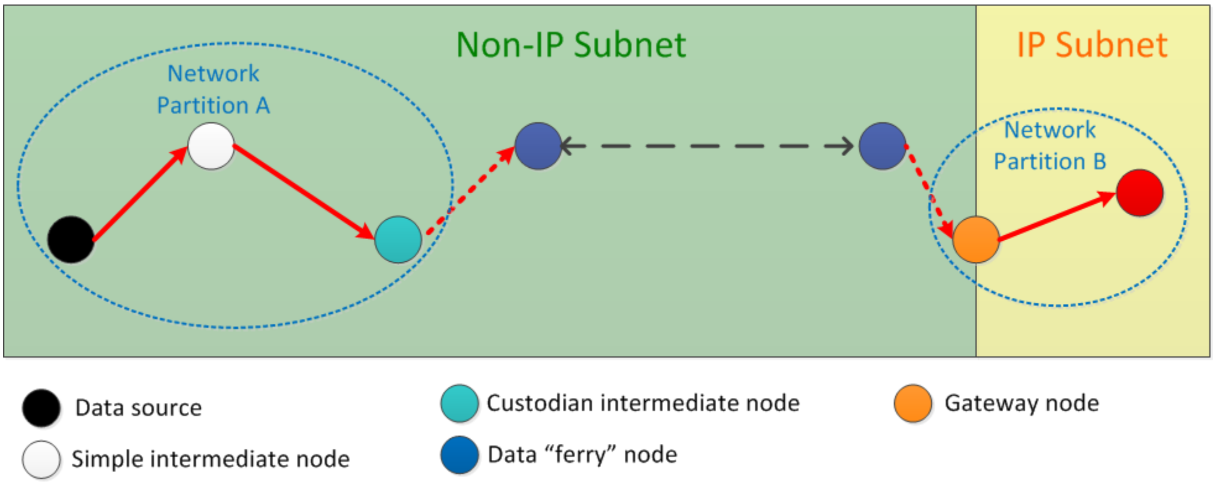

Figure 2.1: DTN scenario example

2.2.1 The Bundle and Convergence Layers

The Bundle layer is the basic novelty introduced by the DTN architecture, to support scenarios such as the aforementioned one. It constitutes a new sublayer within the application layer of the protocol stack, as shown in Fig. 2.2, where the majority of DTN operations are placed. These operations include the data storage capability at the intermediate nodes, data fragmentation operations, the hop-by-hop reliability strategy (discussed in section 2.2.2), as well as a general framework for supporting different QoS classes of service [30], [31]. Last but not least, based on the DTN Architecture, the bundle layer can integrate data routing, scheduling and congestion control intelligence within the intermittently connected parts of the network. The bundle protocol specification [30] specifies the "bundles" format. Bundles are data units which are constructed out of the Application Data Units (ADUs) and are generally supposed to be large and self-sufficient (i.e., include both data and all necessary metadata information), in order to avoid "chatty" negotiations with the receiver nodes and comply with the limited amount and duration of node contacts.

Based on the DTN Architecture, the DTN-related functionality implemented at the Bundle layer can be agnostic of the protocol stack lying underneath. To support this feature, the DTN architecture supports the existence of convergence layer adapters. The aim of these adapters is to provide appropriate interfaces to adapt the Bundle Layer's operation and requirements to the services and specification of the protocol stack which is available for each local network. As there can be a large variety of protocol families (e.g. IP vs non-IP, TCP vs User Datagram Protocol (UDP)-oriented for IP links), each respective convergence layer possibly needs to augment these protocols with necessary operations (e.g. message boundaries for TCP streams, reliability, congestion control, segmentation mechanisms for UDP). In this context, the co-ordination of bundle and convergence layer operations is envisioned to ensure the survival of communication disruptions and the interoperability among different network types (as highlighted in Fig. 2.1), respectively, while maintaining the applications running at the communicating ends transparent of the associated mechanisms.

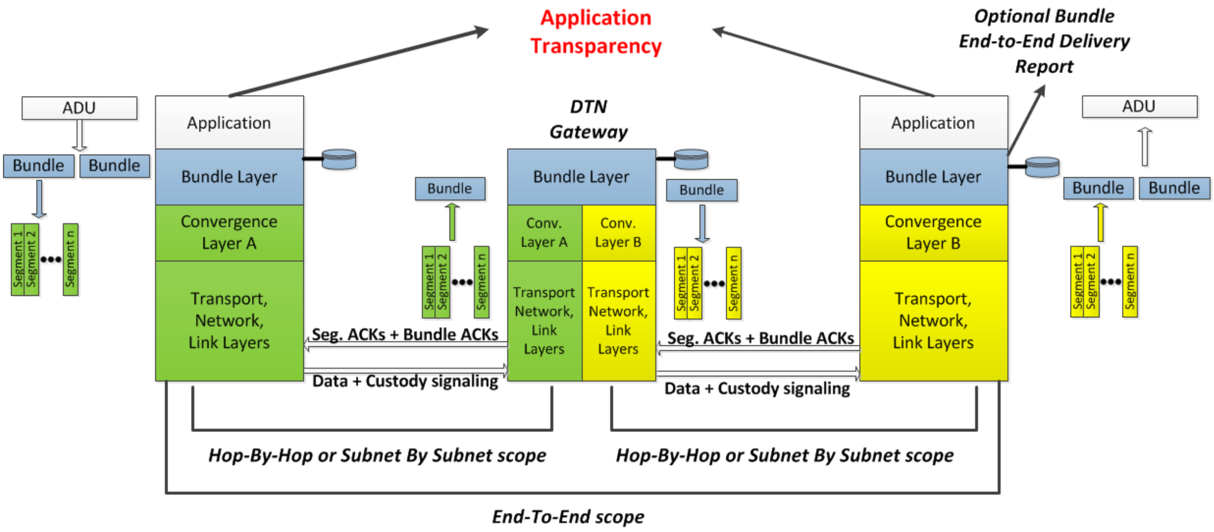

Figure 2.2: DTN architecture overview

2.2.2 Reliability and Custody transfer

Based on the store-carry-and-forward concept, **custody transfer** is the signaling mechanism which supports hop-by-hop (or subnet-by-subnet) reliability, by transferring the responsibility of a Bundle's delivery among the DTN nodes, throughout the path to its destination. This mechanism allows for the reliability operation to move closer to the destination DTN Node, following the transfer of the respective Bundle. Moreover, it allows for fast release of storage resources at the Bundle sources and at intermediate nodes after they transfer the custody to the next hop of the delivery path. Thus, reliability is supposed to cost much less in terms of re-transmissions, delivery delay, data rate and energy consumption. Indeed, considering a bundle transfer failure occurring in some node close to the destination, it would be less costly to re-transfer the failed data from a neighbor node, than from the source node of the initial Bundle. Not all of the nodes throughout a route to the destination have to be custodians and the choice can be based on criteria such as: the amount of available resources that the candidate nodes possess (e.g. buffer space, Energy level) or the topology (e.g. Fig. 2.2 with DTN Gateway being a custodian). Based on [29], a custodian node should normally not be allowed to drop a Bundle under any buffer congestion event.

2.3 Taxonomy of Congestion control and Reliability approaches

In the following, we provide a classification of the congestion control and reliability approaches, existing in the literature. As highlighted in chapter 1, our focus is on solutions for opportunistic contact networks. However, we start from reviewing approaches which target scheduled type of contacts and are more compliant to the DTN Architecture model. The aim is to highlight in which manner such approaches fail to capture the communication challenges in opportunistic networks.

2.3.1 Scheduled vs opportunistic contacts

Inter-planetary and satellite networks are the most characteristic case where the type of contacts between DTN nodes are typically **scheduled**. Due to the resulting predetermined topologies describing them, the main challenge is to deal with the instability of the links within the source to destination path and the large propagation delays, as discussed in section 1. As shown in the example of Fig. 2.1, those links can be grouped in separate network partitions/subnets, residing between data storage points. In this context, within each native subnet, congestion control can be provided in more of a TCP-like manner. To this end, multiple transport and/or convergence layer protocols have been proposed, to provide congestion control and reliability functionalities.

A lot of these are TCP extensions which aim to adjust their congestion control functions, in order to comply with the aforementioned conditions (e.g. [32], [5], [33], [34], [35]). TCPCL [36] is a convergence layer protocol, provided to adopt TCP based protocols operation to the DTN architecture model and the requirements of the Bundle protocol. Other reliable protocols such as DS-TP [37] and NORM [38], [39] aim to apply congestion control in a more efficient manner than TCP, by decoupling the specification of the transmission rate at the sender from the delayed feedback (positive or negative ACKs) arriving from the receiver. Contrary to the aforementioned approaches, LTP [40], [41] and Saratoga [42], [43], [44] focus on point-to-point, data transfer at the link layer. Both protocols support data transport for concurrently running application sessions within the host nodes. Due to their local point-to-point nature and the deterministic means they use to fairly distribute the resources among the application sessions, they do not need to incorporate any congestion control mechanisms, unless interconnection of the links with the public Internet [45], [46] is required. Saratoga supports reliable data transfer, whereas in LTP both reliable and unreliable (UDP-like) data transfer is supported. LTP-T is an extension of LTP to operate in an end-to-end, instead of a point-to-point scope, constituting a multi-hop analog to LTP, as described in [47]. To this end, it integrates reliability through hop-by-hop custody transfer, as well as a simple congestion notification mechanism, based on identifying local storage congestions and notifying the involved peers about it.

Although the aforementioned approaches can tackle reliable data transfer and congestion control within environments with scheduled contacts, the situation changes when contacts become rather **opportunistic**, as in our cases of interest. Then, the main source of intermittent connectivity is nodes random mobility, as described in chapter 1, resulting to topologies which change dynamically and in a non-deterministic way. As a result of these conditions, data transfer decisions have to be taken on a hop-by-hop basis. In this context, it would be rather infeasible to apply congestion control based on the aforementioned end-to-end manner. On the contrary, it is meaningful and important to apply *storage congestion control* at local node buffers. Existing storage congestion control, as well as reliability provision techniques in the literature are closely related to data routing and forwarding decisions. In the context of local decision making, the per bundle related information (e.g., remaining lifetime, priority, size, estimated probability/delay of delivery or further forwarding), as well as the locally (within a node's "neighborhood") experienced storage congestion volumes can be considered to optimize the delivery performance. From an architecture point of view, based on the DTN model, such techniques are usually considered as part of the bundle layer's functionality, as opposed to the previously reviewed transport/convergence layer protocols.

2.3.2 Single copy vs multiple copy routing

The discussion so far has been based on the assumption of a single copy of each bundle existing in the network at each time instant (i.e., **single copy routing**). Indeed, single copy routing and hop-by-hop reliability through custody transfer are generally preferred for scheduled contact networks, which lead to predetermined routing paths. Furthermore, it is often chosen in the context of probabilistic/opportunistic but densely populated networks (e.g., mobile social networks as described in the following), allowing to circulate up-to-date routing information and, at each hop, select the best among multiple relay choices. **Multiple copy routing**, on the other hand, can increase the delivery performance in opportunistic DTN settings. Moreover, it usually requires much less, or even no network topology information [10], comparing to single copy routing approaches. This attribute makes it attractive for a wide range of scenarios where it is hard to circulate and keep such information updated: e.g., sparsely populated networks and stressing communication conditions, on top of nodes mobility. However, multiple copy routing generally comes with the cost of easier resources exhaustion and consequently more frequent buffer congestions, as described in chapter 1. In the context of storage congestion control for opportunistic DTNs, there exist both techniques which are based on single copy (e.g., [48], [49], [50] [51], [52], [53], [54], [55]) and multiple copy routing (e.g., [15] [16], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64] [65], [66]) in the literature.

Among the ones running on top of single-copy routing, only few schemes are independent of the actual routing protocol (e.g. [49], [51]). For instance, Token Based Congestion Control (TBCC) [49] applies some sort of admission control by distributedly controlling the amount of traffic which is injected in the network, resembling the objective of congestion control of TCP-based protocols. On the other hand, multiple approaches, primarily focusing on social opportunistic networks, combine their congestion control and data forwarding strategies with exploiting routing information. Such information may refer to the *congestion level within a node's neighborhood* (e.g. [48], [50], [52], [53]) or some *social metric* which can indicate the relay nodes that lead to faster delivery (e.g. [50] [52], [53], [54]). CaFe [50] is one such approach which combines both these types of information to optimally select the next hop for a bundle's delivery.

Regarding congestion control with multiple copy routing, many popular protocols (e.g., [11], [67], [68]) are based on restricting the amount of replication, in order to decrease the overall congestion in the network, comparing to pure epidemic routing [10], while preserving higher delivery performance than single copy routing, in many scenarios of interest. However, they suggest rather static ways for limiting replication. In this context, determining optimal replication factors for dynamic opportunistic networks is a challenging task. To this end, congestion control techniques, which can dynamically adjust data replication decisions based on the experienced level of congestion in the network, are considered necessary.

A wide range of such existing techniques is *independent of classic routing information related to destination-based*, best relay selections (e.g. [15], [16], [55], [61], [62], [64], [65], [66]). Instead, their resources distribution methods, can be classified based on whether they use some sort of *replication* or *buffer dropping management*, or both, as in [15], [16], [64], [65], [66]. Replication management techniques can be further categorized to those adjusting their replication factors based on the cooperatively experienced congestion volumes at the DTN nodes (i.e., *node-based experienced congestion*) (e.g., [55], [61], [63]) and those that determine the degree of replication on a *per bundle* basis, based on bundle-specific (non-destination related) information (i.e., remaining lifetime, priority, bundle size, number of copies etc.) (e.g., [15], [16], [62], [64], [65], [66]). The former approach is based on some type of storage congestion indications originating from a node itself, or considering also relative information from its neighbor nodes. Based on such indications, each node locally and dynamically determines the degree of replication which is applied on equal terms among the stored data. The latter approach is considered more attractive for our QoS prioritization framework, which dictates the need for replication and dropping decisions in different terms, depending on the relative priority among distinct application sessions. Global Knowledge Based Scheduling and Drop (GBSD) and History Based Scheduling and Drop (HBSD) [16] rely on both bundle-based replication management (scheduling) and dropping, to optimize resources distribution during limited contact durations and buffer congestion events, respectively. Particularly, these policies are based on deriving per bundle utilities which express each bundle's marginal value, with respect to the network's optimization metric of interest (i.e., delivery rate maximization, or delivery delay minimization). Moreover, they operate on top of epidemic routing. Without excluding the use of other multi-copy routing protocols, selecting greedy epidemic routing underlines more the performance value of the specific approach, since it demonstrates its efficiency under the most stressing conditions. In [64] a variant of the GBSD and HBSD schemes is proposed based on the same framework of per message utilities but designed to operate on top of binary Spray and Wait, instead of Epidemic routing.

2.3.3 Congestion control objective

In the context of single copy routing and hop-by-hop custody transfer, although buffer congestions are expected to be less frequent than with multiple copy routing, their effects are relatively more detrimental for the network performance, than with multiple copy routing. Indeed, if a node has to drop a bundle for which it has accepted the custody, then this bundle will surely not be delivered, since there will be no way for the source node to get informed and re-transmit it. Hence, multiple storage congestion control techniques on top of single copy routing mainly aim to *avoid congestion events* (e.g., [48], [49], [50], [52], [53]).

On the contrary, congestion control techniques based on multiple copy routing may refer either to *congestion avoidance* (e.g., [55], [56], [58], [59], [60], [61]), or *congestion management* (e.g., [15], [16], [57], [62], [64], [65]), or both (e.g., [63], [66]) aiming to minimize the negative effects of buffer congestions, once they occur. Since multiple replicas of the same data can co-exist in the network, the impact of dropping bundle copies can be significantly lower, if the copies to get dropped are chosen optimally with respect to some network performance metric(s). Accordingly, the benefits of bundles replication can be maximized, if the bundles to get replicated during limited contact durations are picked optimally, based on criteria as those described before.

2.3.4 Reliability and data acknowledging objective

As already discussed, traditionally reliability refers to the capability of always ensuring successful data delivery at the initial source (end-to-end scope). This capability strongly depends on the utilized acknowledging mechanism (ACKs). The ACKs are usually short control packets which traverse the network on the reverse path, with the aim of reaching the initial packet source and inform it about the successful delivery of the respective data packet at the destination. In DTNs, however, the acknowledgments aim to provide reliability either on a **hop-by-hop** (subnet-by-subnet) basis, or on an **end-to-end** basis. Although the former consists the basic alternative to traditional end-to-end reliability for DTNs [29], [69], its efficiency is questionable for non-scheduled contact scenarios which might lead to unexpected bundle drops at custodian nodes. STRAP [70] is based on custody delegation to provide hop-by-hop reliability to multicast opportunistic networks. In this context, it requires to maintain per bundle delivery state information at the DTN nodes and circulate it in the network. Hop-by-hop custody transfer is also not appealing to combine with multiple copy routing schemes, due to the increased complexity and associated overhead of keeping track of multiple paths and/or sub-paths, as discussed in chapter 1. End-to-end reliability approaches, on the other hand, are always challenging to provide, due to the absence of end-to-end connectivity, let alone in the framework of opportunistic scenarios. In this context, some of the existing approaches provide **best effort mechanisms for end-to-end reliability** (e.g., [71], [72], [73], [74]), while others intend to **guarantee end-to-end reliability** (e.g., [75], [76], [77]). The main differentiation point among the two categories has to do with how each one performs when delivery time limits are imposed. Although best effort approaches can ensure 100% delivery ratio when there are no time restrictions imposed on the data delivery, they cannot do the same when such restrictions are present [71]. The aim of guaranteed approaches, on the other hand, is to ensure this ratio even under time constraints.

To this end, the aforementioned guaranteed schemes combine network coding with ACK mechanisms. Network coding generally allows to encode and merge multiple individual packets in a single one, permitting to increase the amount of data that flows in the network and decrease the required resources per packet. As a result, its use is quite popular with DTN solutions. Ali et al. [75] are based on retransmission cycles to guarantee end-to-end reliability, while the use of Random Linear Combinations (RLCs) of individual packets assists in reducing the amount of retransmitted data and minimizing the overall data transfer time. In [76] they extend their approach to account for both unicast and multicast delivery, as well as for supporting multiple sessions launched concurrently in the network. Such approaches seem promising for guaranteeing end-to-end reliability in opportunistic settings. However, the fact that their evaluation is based on a relatively small number of individual sessions running between source-destination pairs, raises some scalability concerns about how they would respond to a larger amount of concurrent sessions in the network. Accordingly, from a mobility perspective, they lack some assessment with real rather than synthetic mobility, to better validate their performance in opportunistic settings.

Reliability is not the only role of acknowledging mechanisms. ACKs can also be used as means of closed loop congestion control, by **releasing network resources** (e.g., [71], [74], [78]) which are attributed to bundles that have already been delivered (e.g., buffer space, redundant future bandwidth/energy consumption for replicating such packets). This role is more important when multiple copy routing schemes are used and, as a result, the amount of utilized resources per packet in the network is much larger than in the case of single copy routing.

2.3.5 ACK dissemination scheme

However, given the limited amount of resources and communication opportunities, the dissemination of data ACKs can consist a significant source of overhead in the network. In this context, existing schemes intending to capture one or more of the aforementioned reliability objectives, incorporate different ACK dissemination approaches.

In [71] Harras et al. suggest some basic alternatives, aiming to operate on top of multiple copy data routing. Active and Passive receipt are two best effort reliability and resources releasing schemes, based on spreading multiple copies of each ACK in the network. The two approaches differ in the "aggressiveness" in which they spread the ACKs. Active Receipt is based on **Epidemic routing** (as [72], [73], [75]) and thus induces more control traffic in the network comparing to Passive receipt. The latter disseminates the ACKs only to nodes which are "infected" with the respective data that ACKs are targeting, thus constituting a **selective** way of replication. This however comes with the trade-off of increased queueing time of the initial data at the DTN node buffers, comparing to Active Receipt. In an attempt to balance this trade-off, Congestion Level based end-to-end ACKnowledgement (CL-ACK) was proposed as an extension of the aforementioned approaches, which switches dynamically between Active and Passive receipt, based on the measured congestion level (i.e. message drops/message replications). Another technique used to increase the efficiency of feedback dissemiation is **ACKs aggregation** (e.g., [76], [77]): multiple ACKs aggregated to single messages, as a means of reducing the overhead of utilized resources and achieving faster spreading. In [76] the following aggregation attributes are used: the inclusion of multiple ACKs to a single Selective ACK (SACK) which acknowledges the reception of multiple messages from different sources at a specific destination; The Global Selective ACK (G-SACK) which is produced by merging the contents of multiple SACKs traversing the Network.

2.3.6 Considering multiple QoS classes

Multiple of the previously reviewed schemes aim to optimize resources distribution in the context of congestion control and reliability provision. However, they generally consider application sessions of equal priority (i.e., of a single QoS class). As discussed in chapter 1, though, our aim was to come up with appropriate prioritization policies which would "add the dimension" of multiple QoS classes support, in the framework of optimal resources distribution.

Based on the bundle protocol [30], there is provision for three different QoS classes: Expedited (high priority), Normal (medium priority) and Bulk (low priority) by the DTN community [1]. More recently there has been an extension to support more priority levels within the Expedited class [31]. While such QoS classes provide a static characterization of different classes of messages, prioritization decisions among bundles belonging to different classes is an open issue. In this context, a few existing schemes claiming to address the problem of QoS prioritization, do so by reserving a fixed proportion of resources to each class based on its relative priority comparing to the others (e.g. [79], [80]). As it will be analytically discussed in chapter 4, though, such static approaches cannot comply with dynamically changing opportunistic DTN environments. Similarly, other existing approaches are rather heuristics (e.g., [81]) which would also fail to guarantee the intended, resources-aware, performance, as this was described in chapter 1 (section 1.3).

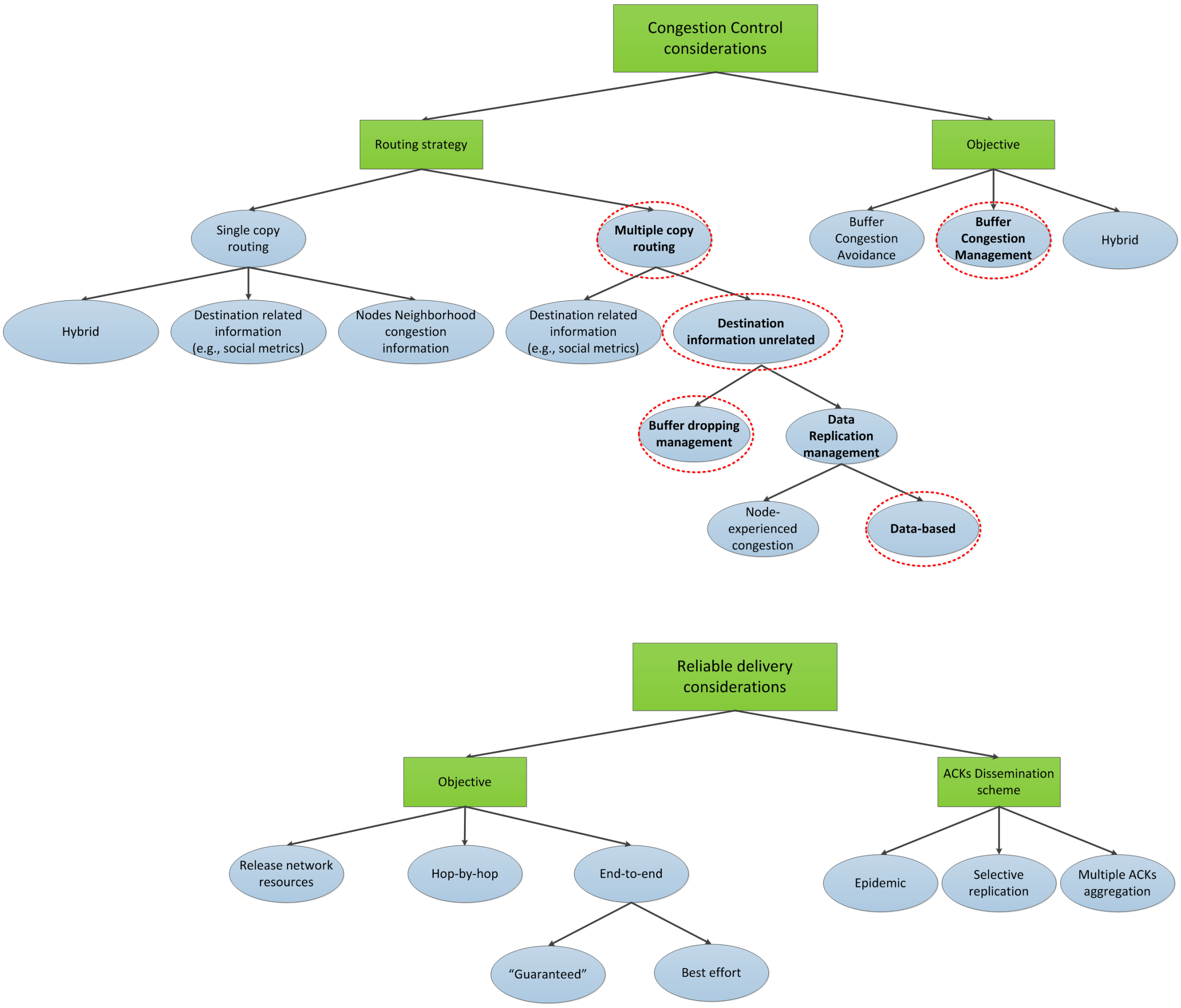

Figure 2.3: Taxonomy of congestion control and reliability considerations for opportunistic DTNs, highlighting our design choices for data QoS prioritization, in red

2.4 Summary of design choices

Based on the review of the aforementioned approaches and their respective taxonomy (summary in Fig. 2.3), we highlight here the basic design choices, with respect to our QoS prioritization policies:

- Due to the opportunistic nature of the environments of interest, we chose to build our approach on top of *multiple copy routing*. Although, among the reviewed approaches for opportunistic networks (section 2.3.2), a great proportion of them has been based on single copy routing, we claim that such approaches mainly focus on social networks: i.e., dense networks where multiple candidate relays can co-exist for every piece of data, allowing to select the best of them, based on multiple routing criteria. However, in our cases of interest (chapter 3), on the one hand, this high density characteristic might not be the case (e.g., tactical networks) and, on the other hand, any relevant up-to-date routing information might be more difficult to disseminate comparing to social networks (e.g., tactical, vehicular environments). To this end, multiple copy routing is a more reasonable choice, since it provides a framework to better exploit the limited communication opportunities and operate without, or with minimal routing information, comparing to single copy routing.

- In this context, our approach towards congestion control relies on *replication* and *buffer congestion management* on a per bundle basis. As it will be described analytically in chapter 4, this approach depends on making accurate per bundle delivery predictions and respecting the sequence of QoS class delivery requirements, to determine the bundles prioritization order, whenever limited duration contacts or buffer congestion events occur.

- Based on this delivery predictions framework, we aim to "guarantee" the per class QoS requirements on average, whenever this is feasible based on the storage resources availability, without accounting on any sort of acknowledges-based reliability scheme, as the ones described previously. In this manner, we avoid introducing any ACKs propagation and storage overhead. However, we claim that some ACK dissemination scheme could be integrated in our policy, to assist in faster resources release and potentially improve the overall delivery performance. For a given amount of available resources, though, such an improvement would depend on how well the trade-off between the ACKs storage overhead and the gain from ACKs dissemination, as a congestion avoidance mechanism, is balanced.

Chapter 3

Use case scenarios

3.1 Introduction

As discussed in chapter 1, we envision our QoS prioritization framework to be applicable for a wide range of opportunistic DTNs. In the current chapter, we focus on the two use case scenarios, introduced in chapter 1 and which we have reviewed in some greater detail. Our aim is first to indicate the importance of concurrent QoS provision for distinct application classes in these scenarios. Then, through the description of the respective networking environments, their significant differences are highlighted. Yet, for each scenario we specify the context in which the delivery performance can leverage from our distributed QoS framework.

One of the differences between the two scenarios is with respect to the capability for direct, non-DTN based communication. In the military scenario such a capability is not considered, whereas in the Floating Car Data upload use case, the DTN functionality is considered as an alternative, only when direct delivery is not possible. Thus, an hybrid functionality is required, which supports the transitions between the two delivery modes, according to the experienced networking conditions and the nature of the corresponding application sessions. Although we highlight some cases where such transitions are required (e.g., data traffic congestions at the upload targeting cells), further transition details were out of our scope. Moreover, from a mobility perspective, we draw the attention on the fact that, in the military use case, homogeneous type of contact patterns are considered, as opposed to the FCD use case where they are more likely to be heterogeneous.

From a practical implementation perspective, we suggest a corresponding architecture for every use case, appropriate to support its operation within the required context. Thus, for the military use case, we are based on the Bundle protocol [30] and ProPHET [67] specifications to suggest the necessary extension modules which can support our QoS solution framework. For the FCD upload use case, on the other hand, we use the ETSI ITS architecture [82] as a basis on top of which we suggest the placement of appropriate blocks to support the transition capability between the two delivery modes, as well as our DTN based functionality.

The rest of this chapter is organized as follows. In sections 3.2 and 3.3 we analyze the scenario specific considerations, including the respective QoS application concepts, the mobility scenarios and the corresponding architectures. In section 3.4, we conclude the chapter.

3.2 Military tactical scenario

3.2.1 QoS concepts in Military Networks

As discussed in chapter 1 (section 1.2.1.1), multiple types of applications should be supported concurrently in a military context (Fig. 3.1).

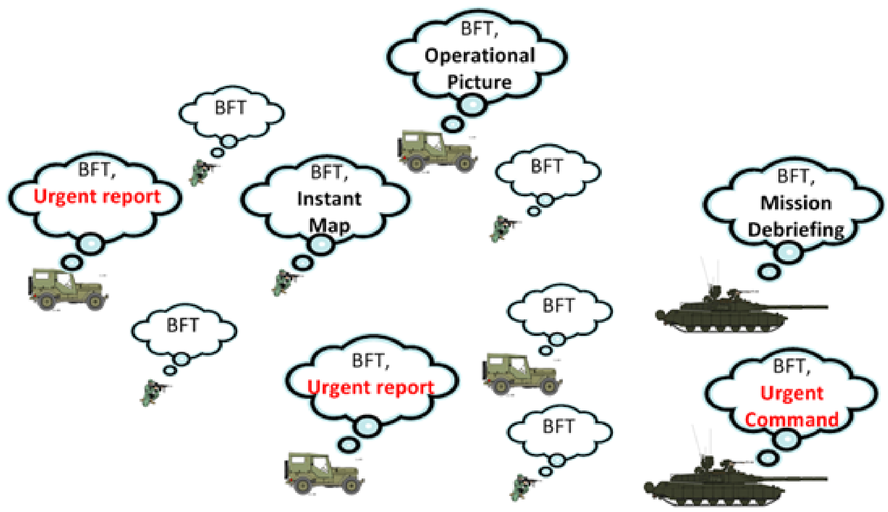

Figure 3.1: Support for multiple competing applications

Among the data traffic types considered in MIDNET project, the following classification can be made with respect to their objective, as described in [17]:

- **Situational awareness** traffic typically refers to periodic position updates (e.g., Blue Force Tracking (BFT) service, reporting the position of friendly forces in the battlefield). If such information is out of date, it is usually not useful to the recipient. As a result, such type of traffic has usually short delivery delay requirement. On the other hand, more recent updates from a source node are expected to obsolete the older ones, allowing the latter's removal from the buffer of the receiver side. Moreover, considering the periodicity in the generation of position updates, it can be assumed that some amount of delivery failures can be tolerated (i.e., relatively low delivery ratio requirements comparing to other services). Apart from positioning information, situational awareness traffic may also refer to maps, battlefield conditions, as well as text and image messages.

- **Reports** can refer to both high and low priority information. Thus, low-priority reports may be attributed to vehicle status reports or mission debriefings, while an example of high-priority report would be a vehicle running low on ammunition. While some types of reports may not have short delivery delay requirements, their delivery should be ensured with a high probability (i.e. high delivery ratio requirement). As the traffic is very varied, it is hard to deduce any general requirements, apart from the fact that it should be assigned with different nominal priorities.

Based on the aforementioned requirements per traffic type, the generic categorization depicted in figure 3.2 was made in [17] to represent the relative priorities between them. In this context, retention priority refers to the criticality of maintaining the respective data stored at the DTN nodes (i.e., corresponding to delivery ratio requirements); transmission priority captures the importance of fast delivery (i.e., corresponding to delivery delay requirements).

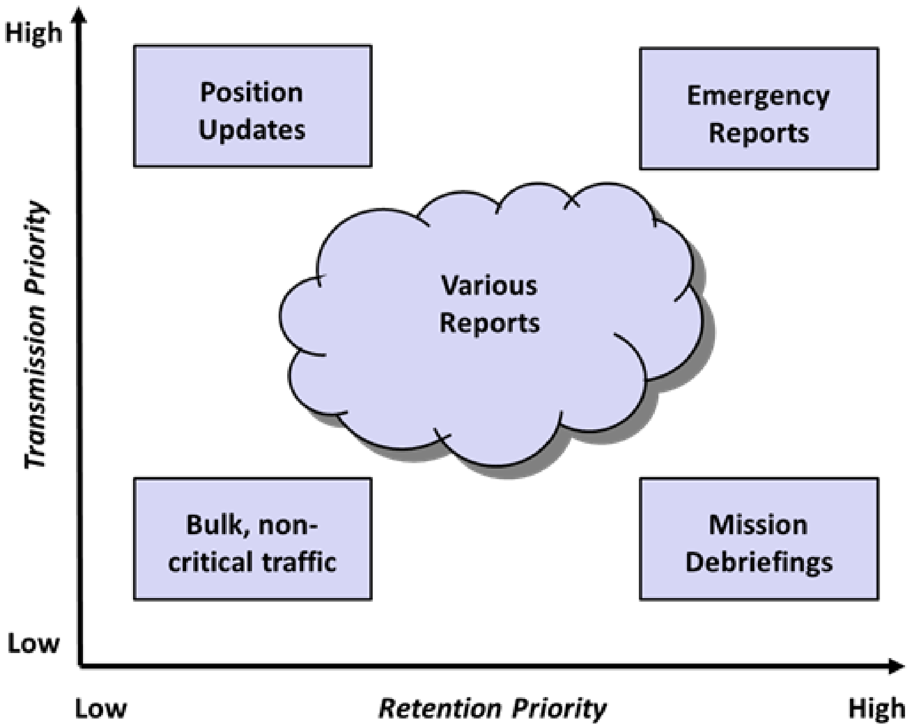

Figure 3.2: Traffic type examples with different transmission and retention priorities

Under these considerations, multiple transmission and retention priority classes were defined to support the applications framework in MIDNET [17]. In resource constrained environments (i.e. Bandwidth, contact opportunities, storage), such classes can constitute indicators for prioritization decisions. In this context, a QoS prioritization framework was envisioned to operate in a two-level manner. The first level refers to static policies which will “filter” the messages which should in any case be prioritized or dropped first. Examples of such policies consider the following criteria:

- **Updates:** which when available are allowed to delete older messages of the same application (e.g. BFT as discussed before).
- **Very urgent messages (highest priority):** which should, in any case, be replicated first.
- **Bundles destination:** During node meetings, the bundle copies whose destination is the encountered node should get high scheduling priority, to ensure that they won’t miss the delivery opportunity, regardless of their nominal QoS class.

The second level refers to dynamic policies which aim to avoid the starvation of lower priority classes, while respecting the QoS standards of higher priority classes in resource constrained environments, as discussed in chapter 1. This is where the distributed QoS scheduling and buffer management schemes presented analytically in the next chapters come into the picture.

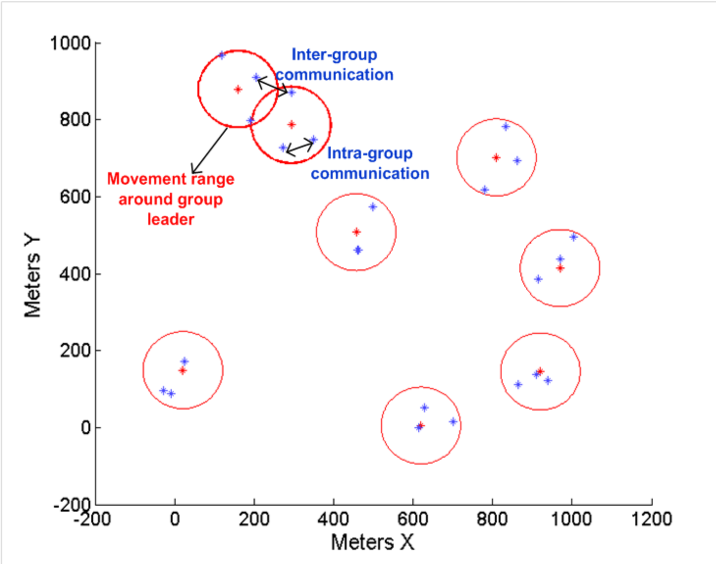

Figure 3.3: Group based random waypoint mobility model example for open terrain scenario

3.2.2 Mobility scenarios

The applicability of our QoS prioritization scheme was envisioned in the context of open terrain mobility scenarios. As described in [83], an open terrain in a battlefield refers to a large, mainly flat geographical area, without obstructions such as buildings and trees. Such terrain is often used for tactical operations, as the lack of obstacles makes mobility conditions easier, allowing fast and long distance movement of mainly motorized forces.

In this framework, we considered a number of distinct squads (groups) patrolling an area of specific size. Each group consists of a leader and subordinate nodes following it. Based on the adopted mobility model for such scenarios in MIDNET, the leader node selects trajectories based on the random waypoint mobility model [84]. The model's configuration includes parameters such as the value ranges of the uniformly distributed speed and pause times (i.e., time intervals during which a node stops moving after having reached the end of a trajectory and before selecting a new one). The rest of the group nodes follow the selected trajectories by moving randomly around their leader in an area of restricted radius. In Fig. 3.3 a snapshot of this group-based random waypoint mobility is depicted. As you can see, each group is composed by a group leader (red node) and three follower nodes (blue nodes).

In this context, QoS prioritization policies were envisioned to operate both within each group's scope (i.e., intra-group communication), as well as within nodes belonging to different groups (i.e., inter-group communication): e.g., the leader of one group needs to get data delivered to the leader of another distant group. In any of the two scenarios, the subordinate nodes of the same or different groups can be used as relays by means of data replication, when they come within communication range, as shown in Fig. 3.3. Based on the aforementioned mobility scenario, the inter-contact patterns among different node pairs having the same "group-based relationship" (i.e., either belonging to the same, or different groups) are expected to be **homogeneous.**

3.2.3 Suggested QoS supporting Architecture

In figure 3.4, a DTN-based architecture integrating the QoS-related routing modules is depicted. This architecture [85] relies on the model proposed by the DTNRG community [29], which was reviewed in chapter 2, combining functional blocks introduced by the Bundle protocol [30] (i.e. bundle protocol agent) and the PRoPHET [67] routing protocol (i.e., routing protocol agent and neighbor discovery block). PRoPHET is a multiple-copy routing protocol, which was used and extended in the context of the MIDNET project. Although our QoS based scheduling and buffer management policies are independent of any PRoPHET-specific routing functionality, we claim that its suggested building blocks and their associated interfaces can host our schemes intelligence, together with some basic supporting mechanisms (e.g., neighbor discovery), as it will be described in the following. Moreover, these blocks consisted the basis on top of which other types of routing policies were developped in the context of Midnet project (i.e, resource and geographic based routing), thus supporting this architectural pattern for consistency reasons. In the following, we highlight the envisioned QoS routing functionalities within each block.

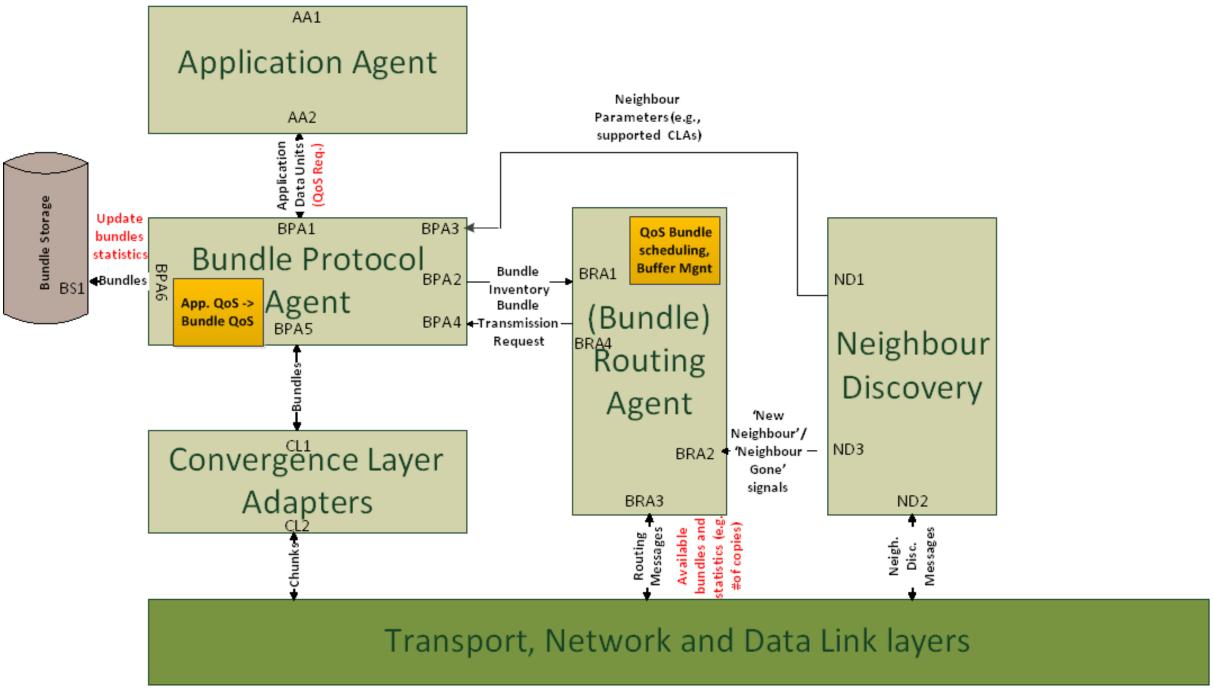

Figure 3.4: QoS-based MIDNET architecture

- **Bundle Protocol Agent (BPA):** This block integrates the main functionality of the Bundle protocol [30], providing appropriate interfaces with the different applications on the one hand and with the convergence layer adopters on the other, to support bundle transfers during node encounters. In this context, a sub-block (**App. QoS → Bundle QoS**) is introduced, which is responsible for translating the ADUs QoS requirements (i.e., minimum delivery ratio and maximum delivery delay in our case) into appropriate classification parameters which can be integrated within the primary block [30] and the Extended Class Of Service (ECOS) [31] block of each Bundle. Moreover, the BPA provides access to the local bundle storage of each node, through a corresponding interface, responsible for storing and retrieving bundles, as well as updating bundles related statistics, which will be explained later.

- **Bundle Routing Agent (BRA):** This block is envisioned to integrate the intelligence of our QoS prioritization framework. In this context, it interfaces with the neighbor discovery block (*ND3-BR2*) to first get informed about the upcoming contact opportunities. Through the *BPA2-BRA1* interface with the BPA, it retrieves the list of available bundles, plus any relevant bundle dissemination statistics (e.g., estimations on the number of copies per bundle) from the local buffer. On the reverse side the same interface can be used to update such statistics based on the information obtained from the encountered nodes. The aggregated control information is forwarded to the networking stack for transmission through the *BRA3* interface.
  - **QoS scheduling and buffer management** sub-block: Based on the received list of available bundles at the encountered node and the aforementioned dissemination statistics, this module is responsible for specifying the replication (scheduling) sequence, among the non-common bundles with the other node. This sequence is then send to the BPA in the form of a transmission request, through the *BRA4-BPA4* interface. In the case of congestion events, the buffer management module is based on locally available bundle dissemination information to determine the bundles dropping sequence.
- **Neighbor discovery:** In the framework of our QoS policies, only basic neighbor discovery mechanisms (e.g., based on periodic beaconing) are required and assumed, to trigger the bundle replication procedures, as discussed previously.

3.3 FCD upload in hybrid vehicular networks scenario

3.3.1 Traffic offloading and QoS considerations

As discussed in chapter 1 (section 1.2.1.2), Floating Car Data (FCD) represent a versatile source of data traffic, originating from moving vehicles. Such data can refer to collecting vehicles geo-location information for the sake of traffic management applications, urban sensing (e.g., reporting pollution levels or mobility patterns), or vehicles distant sensor monitoring, useful for the car manufacturers (e.g., diagnostics, anti-theft operations) [20]. With the increasing amount of vehicles or smartphones generating FCD, as well as the competition with urgent type of traffic (e.g., safety applications), it is not hard to realize the emerging challenges in terms of data traffic load management at the edge of the Infrastructure Network (i.e. RSUs or Cellular Base Stations). To this end, the DTN framework can be used as a means of temporal FCD traffic offloading, through its capability for intermediate data storage, which would permit the data transfer to the infrastructure network, only when the data traffic conditions allow to do so (Fig. 3.5).

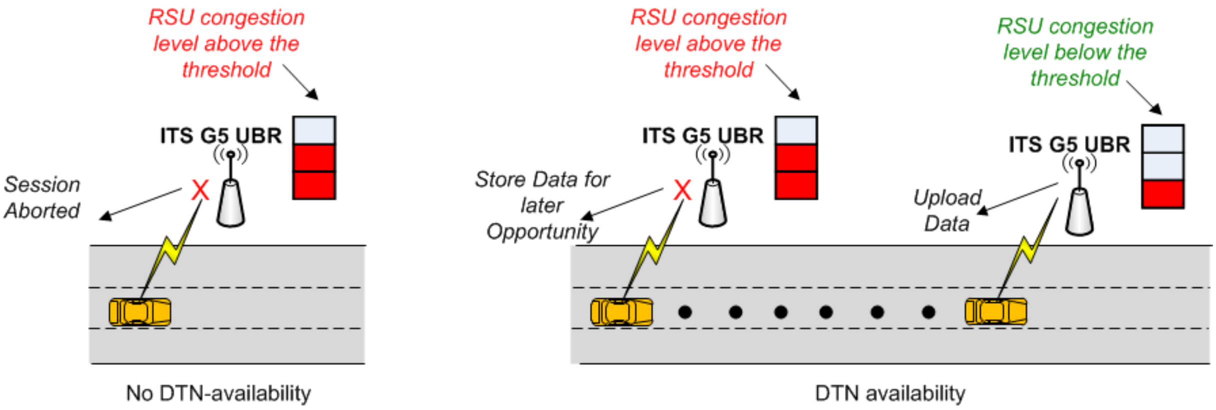

Figure 3.5: DTN vs non-DTN availability

In this context, although the DTN solutions generally target non-real time applications, different FCD application families should be expected to have different QoS requirements. For example, similarly to the use case of military networks, we could consider that traffic management data could have stricter delay requirements but, given their envisioned periodic generation, less strict delivery ratio requirements comparing to aggregated car sensor statistics data, for example. Consequently, it is also meaningful here to consider methods for QoS prioritization among different QoS classes of service.

In the following, we first highlight some mobility examples of interest with respect to the associated communication disruption scenarios. Then, we suggest generic extensions on top of the standard ETSI architecture for Intelligent Transportation Systems (ITS) [82], in order to provide an hybrid functionality which supports the dynamic switching between DTN and non-DTN (direct delivery) mode, according to the resources and communication opportunities availability. Based on this architecture we propose a placement of the appropriate DTN functionality blocks, integrating our QoS prioritization policies.

3.3.2 Mobility and resources related considerations

In contrast to the described military use case (section 3.2.2), where the objective was to deliver data traffic to mobile nodes, here the aim is to upload FCD traffic through some smaller or larger coverage cell, residing at the network edge. In this context, FCD transfer might not be possible at the moment it is available for uploading due to the cell being congested, as discussed previously, or due to the short contact time of the vehicle carrier with the cell, comparing to the amount of FCD that needs to be uploaded. Such conditions can trigger the switch to DTN delivery mode for some selected data sessions.

Within the DTN mode, the combination of our prediction based, QoS prioritization framework with data replication can be strongly dependent on the description of the pairwise intercontact patterns between the vehicles (V2V scope), or between the vehicles and the infrastructure nodes (V2I scope.). In this context, it should be expected that, as opposed to the aforementioned military mobility scenario (section 3.2.2), these patterns should expected to be rather **heterogeneous** in real world mobility scenarios.

3.3.3 DTN and QoS supporting architecture extensions

ETSI Architecture provides the facilities layer which is placed between the Application and the Networking and Transport Layers and integrates functionalities relevant to Application, Information and Session support, as depicted in Fig. 3.6. As described in [82], part of the Application and Session support is to provide applications using backend services with features such as:

- Establishing sessions with the backend,
- Handling unexpected session losses due to the mobility of the ITS station,
- Maintaining a session during handovers.

We claim that the nature of these features provided by the Facilities Layer is close to the functionality framework that we want to integrate regarding:

- The selection criteria to decide among the application sessions which should use Direct Delivery and those which should use the DTN stack instead (when there is a need to do so).
- The QoS-based prioritization decisions among sessions using the DTN stack, under limited resources availability.
- The switching mechanisms which should provide the capability of a session's data transfer dynamically changing to DTN or Direct Delivery operation mode when there is a need to do so. For this function, cross layer information that can be provided through the management entity (Fig. 3.6) should be exploited. Such information is relevant to providing the congestion level at the residing RSUs.

Thus, the Facilities Layer has been selected to integrate the DTN mode selection capability and the DTN related functionality, as an additional way of session handling during any type of disruption event. In Fig. 3.6, we highlight the placement of DTN integrations within the ETSI Architecture model, related to FCD Applications management, as described in [86].

- **Decentralized Congestion Control (DCC) Facility and Flow control Facility:** Those two entities are part of the Management Plane and they are responsible for carrying lower layers information (Access and Transport layer respectively) as congestion indicators. This information is exploited at the Facilities layer within the DTN/ Direct Delivery dynamic Management block.
- **DTN/ Direct delivery dynamic Management:** This block integrates the decision plane of the hybrid functionality. Particularly, based on the knowledge of the available resources, as well as the total amount of data (FCD + rest of Applications) to be uploaded at the core Network, each DTN node (vehicle) has to take a decision regarding which flows will be delivered directly and which through the DTN stack.
- **DTN stack:** All the DTN related functionality is integrated in this block, including the QoS scheduling and buffer management policies. To this end, the block's elements can be structured and interact in a similar manner, as the one described for the military use case scenario (section 3.2.3).

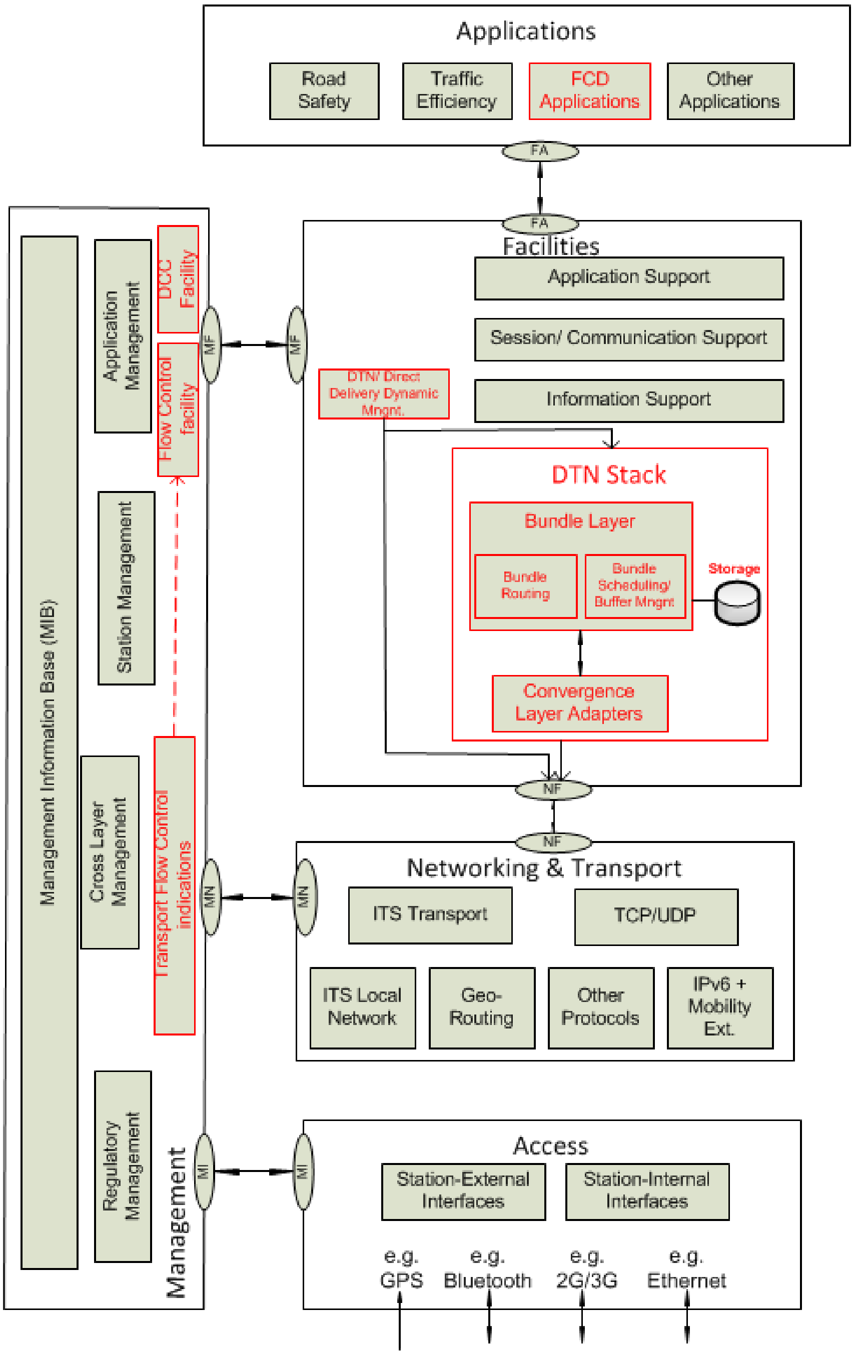

Figure 3.6: ETSI ITS architecture extensions to support DTN and QoS desired functionality

3.4 Conclusions

In this chapter, we reviewed two use cases of interest with respect to QoS prioritization aspects in resource constrained opportunistic settings. The aim of this review was not to restrict the applicability of our QoS prioritization framework to these two scenarios, but rather to highlight its appropriateness for use cases which can differ significantly both in terms of their application context, as well as the associated mobility context. In terms of the latter, we highlighted the differentiation with respect to the inter-contact patterns (i.e., homogeneous in military scenario vs heterogeneous in urban vehicular mobility). As we have indicated also in chapter 1, this is a focal point that is considered by our QoS scheduling and buffer management schemes in chapter 5 and appendix A. Finally, for each use case we suggested appropriate extensions to existing architecture models, to allow hosting the functionality of our distributed schemes.

Chapter 4

Joint distributed scheduling and buffer management policies for DTN Applications of different traffic classes

4.1 Introduction

A lot of multiple copy routing protocols incorporate scheduling and buffer management policies, in order to determine which data should be replicated during a meeting with another node and which data should be dropped during a buffer congestion event [62], [15], [68]. Such policies aim to optimize scheduling and dropping decisions, by keeping track of important message parameters (e.g., number of replicas, remaining TTL) and using them to estimate the probability of encountering the destination. Nevertheless, these schemes assume all end-to-end sessions (and, thus, messages) to be of equal importance.

As highlighted in chapters 1 and 3 this is generally not true. In many envisioned scenarios, network nodes might be running multiple applications in parallel. Although the application framework that we focus on is tolerant to delays, ensuring successful data delivery and/or minimizing the delivery delay may be more important for one DTN application than for another. Consider the military use case example described in chapter 3 where we have two applications launched concurrently at the DTN nodes: one reporting position information of friendly forces periodically and another one generating mission debriefings less frequently. As highlighted in chapter 3 the delivery delay requirement for the first one is lower than the second one, since, after some time, a reported position may be stale. On the contrary, ensuring that a single mission debriefing message is delivered successfully may be more important than losing some (out of many) position updates. It is thus reasonable to assume that different messages might have different QoS requirements, and resource allocation decisions should take these into account.

As discussed in chapter 2, the bundle protocol specification [30] and its extensions [31] already provide the framework to support different QoS application classes concurrently. However, it is not clear how prioritization decisions among bundles belonging to different classes should be taken. If we simply prioritize messages based on their QoS class, then applications belonging to lower classes would “starve” (i.e. they would always be the last to be scheduled and the first to get dropped), if resources are limited (which is most often the case).

A number of recent proposals attempts to address prioritization in a more elaborate manner. In the ORWAR protocol [79], spray-and-wait is used for routing [11], but a higher number of copies is assigned to bundles of higher QoS classes. Additionally, when it comes to dropping decisions, the higher QoS classes are always prioritized (dropped last) over the lower ones (assuming bundles of fixed size). Soares et al. [80] propose different queue sizes, proportional to each QoS class's priority. Dropping decisions of each queue are then independent of the others. Hence, if a bundle of class i arrives and the queue for class i is full, a bundle from that queue will be dropped, even if other queues are not full. In terms of scheduling, they propose that the contact window during a communication opportunity can be shared among different classes, again proportionally to their nominal priority (e.g. 60 % of time for expedited class, 30% for Normal and 10% for Bulk). However, there is no described mechanism on determining the dropping or scheduling sequence among bundles of the same queue.

All the aforementioned policies apply prioritization between QoS classes essentially by distributing the available resources (e.g. number of copies per class, available contact window, available buffer space per class) proportionally to the importance of each QoS class. However, this distribution is based on applying *fixed thresholds*. This raises a number of concerns. First, it is not clear how these thresholds could be tuned based on the environment in hand. Second, fixed thresholds cannot keep up with a dynamically changing DTN environment. Finally, depending on the availability of resources and threshold parameters, the behavior of the policy might be qualitatively different. E.g., if resources are not sufficient to satisfy all constraints, such policies still distribute resources proportionally, and might not satisfy the requirement of any class, not even the highest priority one. On the other hand, if resources are plenty, applying fixed thresholds might keep favoring higher classes unnecessarily (since the marginal utility of extra resources becomes small), and restrain lower classes from achieving a high performance.

To this end, the key contributions of this chapter are to: (i) formally define the problem of *optimizing network-wide performance while satisfying individual class QoS constraints* and (ii) to *derive a distributed algorithm for this problem that adapts to the available resources*. We show that the optimal algorithm for the prioritization problem ends up adding appropriate penalty functions to the optimal utilities of the unconstrained problem of [16]. In this aspect, the work closest to ours is [81], that also extends the optimal utilities derived in [16], to support QoS based prioritization. However, their approach is a heuristic additive term which can neither ensure that QoS requirements are met, nor that the resulting allocation of resources leads to optimal network-wide performance, as we will show. We support our analytical findings using extensive simulations on both synthetic and trace-based scenarios.

The rest of this chapter is organized as follows. In Section 4.2 we present our QoS policy. We start from the description of our system model (Section 4.2.1), and then formulate the QoS prioritization problem, as a constrained optimization problem (Section 4.2.2). Finally, we propose a distributed resource allocation policy that is proven to be equivalent to a distributed gradient-ascent implementation of the solution (Section 4.2.4). In section 4.3, we evaluate the performance of our scheme by comparing it with other prioritization schemes. In section 4.4, we conclude this chapter.

4.2 The QoS prioritization policy

4.2.1 Model description

In the following, we provide some details on our system model regarding our assumptions on mobility, data traffic, application QoS requirements and resource constraints, as well as our general framework with respect to data routing, scheduling and buffer management.

Mobility Model The impact of mobility on the delivery performance, and consequently the performance of our scheme, is determined by the distribution of the **pairwise residual inter-contact times:** *time elapsed until the next contact of a pair of nodes starting from a random observation moment.* Indeed, in our framework, residual inter-contact times can model the distribution of time durations between a moment when scheduling or dropping decisions have to be made and the moment of the next meeting with a bundle's destination. In this context, we assume that there are N mobile nodes in the network and each node encounters other nodes according to a random "contact" process. We model the residual inter-contact times of this process through the exponential distribution with a common rate parameter $\tilde{\lambda}$ (i.e., assuming homogeneous contact networks). It has been shown that a number of mobility models and real traces correspond to contact models with approximately exponential tails [21], [22], [23]. To this end, it can be considered that the inter-contact patterns of the group random waypoint mobility model described in chapter 3 are aligned to this contact model.

Data traffic Model We consider each DTN node running C distinct applications concurrently. Each application generates autonomous data units that we will call bundles from now on, to be compliant with the DTN Architecture [29] and the associated bundle protocol [30]. The bundle generation is modeled through the Poisson distribution with mean rate λ_g per node per application class. We denote as $L_k(t)$ the number of distinct bundles of class k at time t. All application bundles have the same fixed size, which cannot be fragmented. Each individual bundle has a unique destination (unicast) and each transmission is considered successful, if it reaches its destination before expiry (i.e., within its Time-to-Live (TTL) interval).

Application QoS model We associate each distinct level of QoS performance (we will call it priority class from now on) with a specific value of the Bundle Delivery Ratio (BDR) (i.e., *Bundles received on time/Bundles sent*) or Bundle Delivery Delay (i.e., *elapsed time since bundle's creation, until one of its copies is delivered to the destination*). From a bundle's perspective, this requirement can be expressed as its minimum accepted delivery probability $P_{QoS}^{(k)}$, or maximum accepted delivery delay $D_{QoS}^{(k)}$, respectively. We note that each bundle can belong to a single QoS class.

Resource Constraints Our prioritization policy applies to DTN settings with limited buffer availability (b slots per node) and contact windows, comparing to the network traffic load. This load originates from bundle generations at the DTN nodes and bundle replications during node meetings. We model the amount of data which can be replicated within a contact window through the Poisson distribution with a rate parameter r_d.

Routing model, scheduling and buffer management framework We consider epidemic routing. When two nodes encounter, they aim to exchange their non-common bundles. If the contact opportunity is limited comparing to the amount of non-common bundles, the scheduling policy determines which bundles will be replicated. If the amount of replicated bundles leads to buffer congestion(s) at the recipient node, the buffer management policy determines which bundles will be dropped.

Notation	Description
N	Number of nodes in the network
$T_i^{(k)}$	Elapsed time since the creation of bundle i belonging to class k
$n_i^{(k)}(T_i)$	Number of copies of bundle i belonging to QoS class k at T_i
$m_i^{(k)}(T_i)$	Number of nodes who have "seen" bundle i belonging to QoS class k at T_i
$R_i^{(k)}$	Remaining Time To Live (TTL) for bundle i belonging to QoS class k
b	Number of buffer slots per node
C	Number of distinct QoS classes
$L_k(t)$	Number of distinct bundles of class k at time t
$P_{QoS}^{(k)}$	Minimum required probability of delivery for bundles of class k
$D_{QoS}^{(k)}$	Maximum accepted delivery delay for bundles of class k
$P_i^{(k)}(T_i)$	Probability of delivery for bundle i belonging to class k at time T_i
$E\big[D_i^{(k)}(T_i)\big]$	Expected delivery delay for bundle i belonging to class k at time T_i
$\tilde{\lambda}$	Mean inter-meeting rate parameter (exponential distribution)
r_d	Rate of exchanged data per contact (poisson distribution)
λ_g	Bundle generation rate per node per application class (poisson distribution)

Table 4.1: Notation

4.2.2 QoS optimization for average delivery rate

As we have already highlighted, a good prioritization policy should: first, make sure that the QoS requirements of different application classes are satisfied; second, it should allocate the remaining resources, if any, in order to maximize the overall performance of the network. In the current section, we first formulate the prioritization problem for average Bundle Delivery Rate (BDR) maximization, given a set of different application QoS requirements that have to be satisfied. Then, we show analytically how we can obtain a distributed solution to this problem. Our formulation is one of a constrained optimization problem in the following form:

$$\max_{n_i^{(k)}} f(\boldsymbol{n}) = \max_{n_i^{(k)}} \sum_{k=1}^{C} \sum_{i^{(k)}=1}^{L_k(t)} P_i^{(k)}(T_i), \tag{4.1}$$

$$g_k(n_i^{(k)}) = P_i^{(k)}(T_i) \geq P_{QoS}^{(k)} \quad \forall i \in \text{class k}, \tag{4.2}$$

$$Nb - \sum_{k=1}^{C} \sum_{i=1}^{L_k(t)} n_i^{(k)} \geq 0 \quad \forall i \in \text{class k}, \tag{4.3}$$

$$N - n_i^{(k)} \geq 0 \quad \forall i \in \text{class k}, \tag{4.4}$$

$$n_i^{(k)} \geq 1 \quad \forall i \in \text{class k}, \tag{4.5}$$

Based on this formulation, the objective function (Eq. 4.1) can be expressed as the sum of the delivery probabilities of each individual bundle ($P_i^{(k)}(T_i)$) over all bundles and all classes. The delivery probability of a bundle i belonging to class k is the probability of one of its $n_i^{(k)}$

copies to encounter its destination before its TTL expires. In other words, it is the probability of one of its copies next meeting with the destination, occurring in less time than the bundle's remaining TTL ($R_i^{(k)}$). Thus, based on our exponential model of pairwise inter-contact times, it can be expressed as follows:

$$P_i^{(k)}(T_i) = 1 - exp(-\tilde{\lambda} n_i^{(k)}(T_i) \cdot R_i), \tag{4.6}$$

The objective function, denoted as $f(\boldsymbol{n})$ (Eq. (4.1)) and expressed through Eq. (4.6), is concave on $n_i^{(k)}$. The first constraint (4.2) expresses the per bundle delivery probability requirement ($P_{QoS}^{(k)}$), depending on which application class (k) it belongs to. This constraint is concave as well. Constraint (4.3) is linear and states that the total number of bundle copies should not exceed the total buffer space in the network (Nb). Constraint (4.4) ensures that each bundle should not have more copies than the total number of nodes (i.e., no node is allowed to have more than one copy). Finally, constraint (4.5) is there to make sure that a bundle should have at least one copy throughout its lifetime (i.e., the source of a bundle is not allowed to drop it before it expires).

Given that $n_i^{(k)} \in \mathbb{N}$, the above problem is an integer non-linear optimization problem, hard to solve optimally. However, we relax this condition, assuming $n_i^{(k)} \in \mathbb{R}^+$. The continuous relaxation of the problem leads to a convex optimization problem; it can be solved analytically using the method of Lagrange multipliers and KKT conditions [87, chapter 5], to derive a vector of $\mathbf{n}^*$ values that is *feasible*, i.e., ensures that the delivery probability of each message is at least as high as its class requirement, and *optimal*, i.e., $f(\mathbf{n}^*) \geq f(\mathbf{n})$ for all feasible $\mathbf{n}$[1].

However, the above solution requires a centralized implementation of bundle copies, which is not feasible, since there is no central entity in DTNs that could control the state of all messages, instantaneously. Instead, each node only has access to its own buffer content. During a contact between two nodes, dropping a bundle from one buffer or copying a bundle to the node encountered will affect a single variable in the allocation vector $\mathbf{n}$. Hence, two nodes encountering each other can compare the bundles they have in their own buffers and make decisions independently of other nodes. The goal of these decisions should be to modify the allocation vector $\mathbf{n}$ towards increasing the objective $f(\mathbf{n})$. If we ignore the set of QoS constraints (Eq. (4.2)), such a distributed solution has been derived in [16], based on the Exponential expression (Eq. (4.6)). There, the objective is differentiated to get the "marginal gain of an extra copy for each message" (referred to as "message utility") which is equal to:

$$U_i(DR) = \left(1 - \frac{m_i^{(k)}(T_i)}{N-1}\right) \cdot \tilde{\lambda} R_i^{(k)} exp(-\tilde{\lambda} n_i^{(k)}(T_i) R_i^{(k)}), \tag{4.7}$$

Note that, $m_i^{(k)}$ term of the above utility function stands for the number of nodes who have "seen" bundle i (i.e., they have obtained one copy of it at some point, regardless of whether they still have it or not). This term is considered by an enhanced objective function (comparing to Eq. (4.1)), used in [16], to account for the probability of one of these $m_i^{(k)}$ nodes being actually the destination of bundle i. If a node ranks all bundles in its buffer according to this utility, and uses it to make drop or scheduling decisions, then during each contact, the improvement in $f(\mathbf{n})$ will be maximal among all feasible directions (a variable $n_i^{(k)}$ cannot change during a contact, if

[1] In practice, one could round these values to the closest integer to get an approximately optimal solution.

message i is not present in the buffer of any of the two meeting nodes); given the concavity of the objective, this method is shown to correspond to a distributed implementation of a gradient ascent algorithm [16].

Nevertheless, the above solution considers a single priority class only and does not provide any QoS guarantees. Our aim is to modify this distributed algorithm, in order to be able to first satisfy the set of constraints in Eq.(4.2), i.e., to find a "feasible" solution to the problem, and then to maximize the performance among all feasible allocations. In the context of constrained optimization problems, gradient ascent algorithms can be modified to include appropriate penalty functions for each violated constraint [88, chapter 23.5]. Thus, an enhanced objective function would have the following form:

$$\phi(\mathbf{n}) = f(\boldsymbol{n}) - \sum_{k=1}^{C} \sum_{i=1}^{L_k(t)} \psi_k(P_{QoS}^{(k)} - P_i^{(k)}(T_i)), \tag{4.8}$$

where $\psi_k(\cdot)$ is a penalty function related to the constraint for bundles of class k. We would like $\psi_k(x) = 0$, when $x < 0$, i.e, no penalty when the predicted delivery probability $P_i^{(k)}(T_i)$ for message i is large enough. However, we would like $\psi_k(x)$ to take very large values when the constraint is not satisfied ($x \geq 0$), imposing a large negative penalty on $\phi(\mathbf{n})$.

Based on this observation, we can maximize the above objective and solve the constrained version of the problem in a distributed manner, by using the following per message utilities:

$$U_i^{(k)}(DR) = U_i(DR) \cdot \left[1 + \max\{0, c_k(P_{QoS}^{(k)} - P_i^{(k)}(T_i))\}\right] \tag{4.9}$$

In other words, the utility of a message is equal to its unconstrained utility U_i of Eq.(4.7), if the predicted delivery probability is above the class requirement. Otherwise, this utility is incremented by a term proportional to the delivery probability deficit. c_k is a very large constant which ensures that the utilities of bundles that do not satisfy their constraint will always be higher than the utilities of bundles that do satisfy them (to ensure convergence to feasible solutions only).

As a result of these utilities, the bundles which are below their desired QoS threshold are always prioritized (i.e., dropped last, scheduled first) over the ones which are above. Furthermore, note that these utilities correspond to differentiating the extended objective of Eq.(4.8) with $\psi_k(\cdot)$ chosen as an appropriately normalized quadratic penalty function. Hence, ranking and handling (e.g. dropping) bundles according to these utilities at every contact, guarantees: (i) eventual convergence to a feasible solution (i.e, satisfying the constraints), if there is one, and (ii) allocating any "extra" resources optimally, i.e., among all feasible allocations delimited by the constraints[2].

[2]We note here that the above penalty function is not unique in achieving these goals. Other functions penalizing low predicted delivery probabilities sharply could suffice to implement a distributed ascent algorithm moving to feasible and better solutions. However, the priority given between constraints when the feasible domain is empty depends on the penalty function choice, as we shall see later.

4.2.3 QoS optimization for average delivery delay

In the following section, we turn our attention to the QoS prioritization problem with respect to minimizing the average delivery delay. Thus, we proceed with the following formulation:

$$\min_{n_i^{(k)}} f(\boldsymbol{n}) = \min_{n_i^{(k)}} \sum_{k=1}^{C} \sum_{i^{(k)}=1}^{L_k(t)} E[D_i^{(k)}(T_i)], \tag{4.10}$$

$$\begin{aligned} g_k(n_i^{(k)}) =& E[D_i^{(k)}(T_i)] \leq D_{QoS}^{(k)}, \\ &\forall i \in \text{class k} : T_i^{(k)} < D_{QoS}^{(k)} \end{aligned} \tag{4.11}$$

Similarly to section 4.2.2, the objective function (Eq. (4.10)) is expressed as the sum of the expected delivery delays over all bundles and classes. Based on the model of exponential inter-contact times, the expected time until one of the $n_i^{(k)}$ copies encounters the destination, considering also the $m_i^{(k)}$ nodes who have "seen" bundle i, can be approximated as:

$$E[D_i^{(k)}(T_i)] = \left(1 - \frac{m_i^{(k)}(T_i)}{N-1}\right) \cdot \left(T_i^{(k)} + \frac{1}{n_i^{(k)}(T_i)\tilde{\lambda}}\right), \tag{4.12}$$

where $D_i^{(k)}$ stands for the delivery delay of bundle i belonging to class k and $T_i^{(k)}$ is the already elapsed time since bundle's i creation. Based on Eq. (4.12), the new objective function is obviously convex on $n_i^{(k)}$. The constraint of Eq. (4.11) expresses the desired average delivery delay requirement ($D_{QoS}^{(k)} \leq TTL$) for bundles of class k and it is also convex on $n_i^{(k)}$. It is important to stress here that the constraint refers only to bundles whose elapsed time since creation is less than the threshold of delivery delay requirement (i.e., $T_i^{(k)} < D_{QoS}^{(k)}$). This makes sense, if we consider that it is pointless to give higher priority to bundles which have already missed their delay target. Furthermore, such a policy would lead to wasting resources against other bundles which still have the possibility of being delivered before $D_{QoS}^{(k)}$. The rest of the constraints are the same with the delivery rate optimization problem (i.e., Eq. (4.3) - (4.5)).

As for the case of the delivery rate metric, the unconstrained message utility function (i.e., without considering the QoS constraints) for the delivery delay is derived in [16], as:

$$U_i(DD) = \left(1 - \frac{m_i^{(k)}(T_i)}{N-1}\right) \frac{1}{n_i^{(k)}(T_i)^2\tilde{\lambda}}, \tag{4.13}$$

where the number of nodes who have seen the bundle, $m_i(T_i)$, are also considered. Following the same approach with the one described in section 4.2.2, appropriate penalty functions can be introduced in an enhanced objective function $\phi(\mathbf{n})$, to penalize each violated constraint:

$$\phi(\mathbf{n}) = f(\boldsymbol{n}) + \sum_{k=1}^{C} \sum_{i=1}^{L_k(t)} c_i(T_i^{(k)}) \psi_k(E[D_i^{(k)}] - D_{QoS}^{(k)}), \tag{4.14}$$

where $c_i(T_i^{(k)}) = 1$ when $T_i^{(k)} < D_{QoS}^{(k)}$ and zero otherwise, so as to determine whether bundle's i constraint violations are still considered or not. The penalty function $\psi_k(x)$, should take very large values when the expected delivery delay is higher than the class's threshold (i.e., $x > 0$) and zero otherwise (i.e., when $x \leq 0$).

According to the aforementioned rule, the distributed solution of this constrained optimization problem can be achieved by using the following form of per bundle utilities, $U_i^{(k)}(DD)$:

$$U_i(DD) \cdot \left[1 + c_i(T_i^{(k)}) \cdot \max\{0, c_k(E[D_i^{(k)}] - D_{QoS}^{(k)})\}\right]. \tag{4.15}$$

Similarly to Eq. (4.9), c_k is a constant large enough to ensure prioritization of bundles that do not satisfy their constraint over bundles that do satisfy it.

4.2.4 Implementation of the scheduling and dropping policies

In the previous section, we have described a distributed QoS algorithm for the two constrained optimization problems in hand, and have provided theoretical support for its convergence to the desired solutions (i.e., optimal either in terms of delivery rate or delivery delay metric, conditionally on satisfying the requirements). Here, we show a simple implementation of this algorithm, and discuss some additional practical issues. Similarly to the previous discussion, the framework of the suggested implementation is the same for both optimization problems and the differentiation between them lies mainly on the distinct objective functions and the respective utilities and QoS thresholds.

We propose that the bundles residing inside a node's buffer (queue) can be separated in two dynamic groups, as shown in Fig. 4.1: the first group contains all bundles whose predicted delivery probability/delay hasn't reached the desired QoS threshold; the second group consists of bundles which have reached their threshold. In the case of delivery delay optimization, the second group includes also the messages whose elapsed time since creation is higher than the desired threshold (i.e., $T_i^{(k)} \geq D_{QoS}^{(k)}$). The bundles of the first group are always prioritized over the bundles of the second group. Ranking among bundles of the same group is based on the classic utility U_i. It is easy to see that the desired QoS message utility of Eq.(4.9) or (4.15) is monotonically decreasing from left to right in the queue of Fig. 4.1, and thus dropping bundles from the right and scheduling from the left of this queue implements the desired policy.

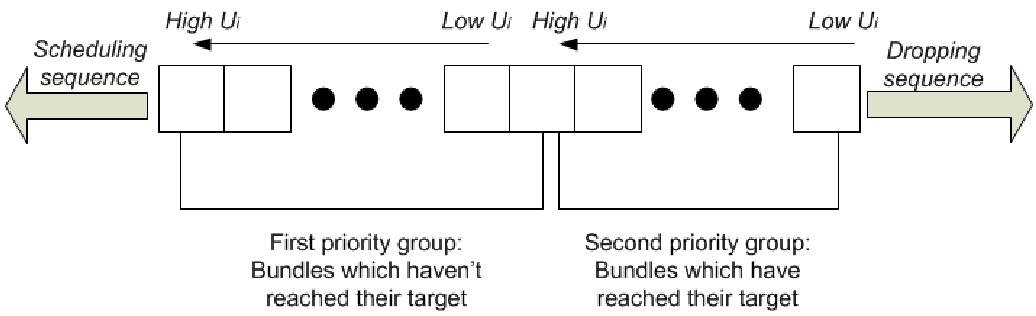

Figure 4.1: Bundle scheduling and dropping sequence

The above policy works fine if the network parameters (e.g. packet generation rate, available storage in the network, inter-meeting rates) permit an algorithm to reach the desired delivery ratios, or average delivery delays, for all priority classes. However, in some scenarios this might not be possible, i.e., the feasible domain of the defined optimization problem is empty. It is somewhat subjective what a desired policy behavior should be, in that case. While one could apply a heuristic ranking in that case, or accept the (infeasible) solution the above policy converges to, in a number of cases we can modify the policy to provably achieve a desired outcome. We believe that an interesting class of cases is when it is more important to try to satisfy the constraints of the higher QoS classes first.

One could achieve this by choosing a different constant c^k in Eq.(4.9) or (4.15), for bundles of different QoS classes (k). Specifically, choosing $c_1 \gg c_2 \gg ... \gg c_K \gg 0$ (with 1 corresponding to the class with the highest nominal priority, and K the class with the lowest one) ensures a "smooth" fallback, if no feasible solution exists [3]. Specifically, if there is a feasible solution, the algorithm converges to it (as explained earlier). But if there is none, it converges to a solution where: for some $j < K$, QoS inequality constraints for all classes from 1 to j are satisfied with equality, class $j+1$ is not satisfied, and all classes larger than $j+1$ (if any) get no more resources than a single copy per bundle (based on Eq. (4.5))[4]. It is important to stress here that this algorithm *does not need to know in advance whether a feasible solution exists.* By construction, it navigates the infeasible domain so as to either enter the feasible domain eventually, or stop at an infeasible solution that is the most desirable one, according to the previous discussion.

The above algorithm can again be mapped into our simple buffer classification system by defining subgroups inside the first priority group (Fig. 4.2): each subgroup is composed of bundles of a particular priority class which are below their threshold (and, in case of delivery delay optimization, haven't missed their QoS target, $D_{QoS}^{(k)}$, yet). In this context, a subgroup attributed to a higher QoS class will always have higher priority than a subgroup of a lower QoS class.

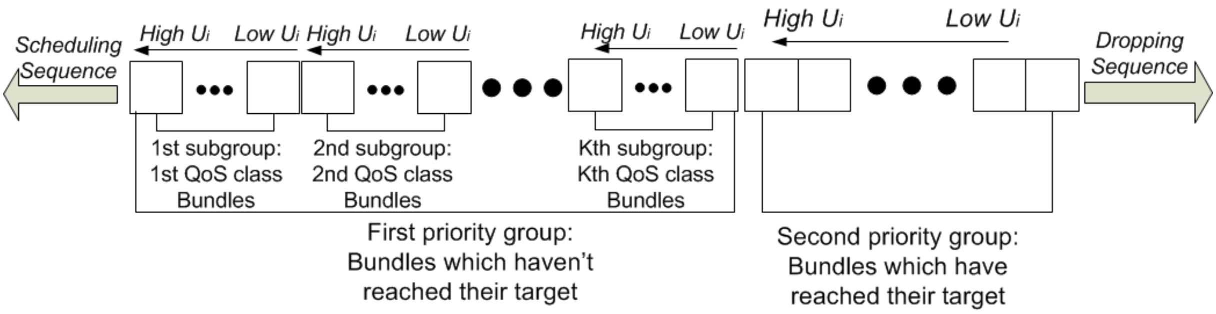

Figure 4.2: An approach for bundle scheduling and dropping sequence for infeasible domains

As a final remark, the above algorithm requires reliable estimates for the number of copies and seen nodes, (i.e., $n_i^{(k)}(T_i)$, $m_i^{(k)}(T_i)$). This is not a trivial problem in a DTN setting. However, it is a problem that has already been addressed efficiently in [16], in the context of the "HBSD" policy. The authors there propose a distributed protocol to obtain estimates of these quantities,

[3] Note that, in practice, it is not necessary to define a QoS threshold for the K^{th} class. Regardless of whether such a constraint is used or not, the corresponding bundles get additional resources, only if they are competing against higher class bundles which are predicted to satisfy their constraints. To this end, it doesn't make any difference if the K^{th} class bundles are classified at the K^{th} subgroup of the first priority group, or at the second priority group

[4] While this starvation of low priority classes is undesirable, when enough resources are available to satisfy all classes, it can be argued that it's a desirable feature in emergency cases with very limited resources. Furthermore, other policies could be defined and achieved by manipulating c_k differently.

and show, using extensive simulations, that the communication overhead is reasonable and that the policy based on the estimate performs closely to one assuming instant knowledge. For this reason, in the remainder of this text we assume that a similar algorithm is implemented, and focus, instead, on the problem of reliably estimating the QoS constraints.

4.3 Performance Evaluation

4.3.1 Evaluation Setup

We evaluate our policy for both optimization problems of interest. In this context, we consider three priority classes, namely Expedited (highest), Normal and Bulk (lowest) (based on the terminology of the bundle protocol specification [30], regarding different QoS classes). The BDR and ADD results are presented for various values of total available buffer space in the network, aiming to test our policy, as we vary the amount of buffer congestions. Equal loads of traffic are generated per application class. The bundles are created following a Poisson distribution with a fixed rate parameter. The scheduling constraints are applied by restricting the average rate of data which can be exchanged per contact to $r_d = 50\%$ of the average unconstrained rate, i.e., the rate of exchanged bundles, in case there is no restriction on the total transferred data per contact. The actual number of exchanged bundles per contact is then drawn from a Poisson distribution with rate parameter r_d. Finally, as highlighted in section 4.2.1 according to our model of synthetic contact traces, the nodes meet each other with a common rate $\tilde{\lambda}$ based on the exponential distribution.

In table 4.2 the scenario configuration parameters are summarized for the exponential synthetic traces.

	BDR Opt.	*ADD Opt.*
Number of Nodes (N)	50	100
Total simulation time (min.)	550	550
Mean nodes inter-meeting rate (λ, min^{-1})	10^{-2}	$3 \cdot 10^{-2}$
Message TTL	1800 sec.	3000 sec.
Mean rate of contact window r_d (% of unconstrained rate)	0.5	0.5
Total bundle generation rate per node (r, min^{-1})	0.08	0.12
Bundle generation rate per node per priority class (r/3, min^{-1})	0.027	0.04
Expedited desired QoS	0.77	400 sec.
Normal desired QoS	0.55	500 sec.
Bulk desired QoS	0.45	-

Table 4.2: Simulation Parameters

4.3.2 Results

Based on our previous descriptions, the intended behavior of our policy is to prioritize bundles in the order of their QoS class importance, when the available resources do not permit to reach the desired performance for all three classes (infeasible domain). In other words, under these circumstances, the first goal is to satisfy the Expedited class, then the Normal class and then

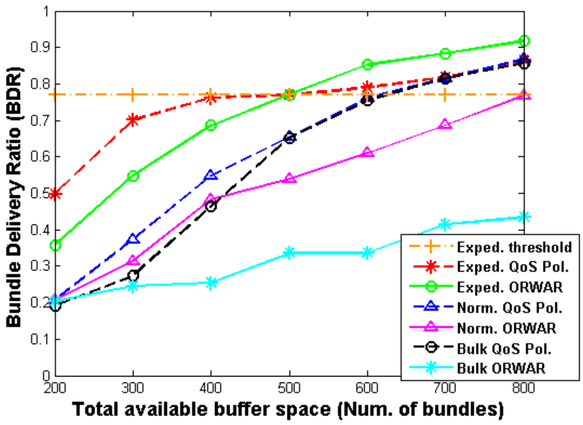

Figure 4.3: QoS Policy vs ORWAR

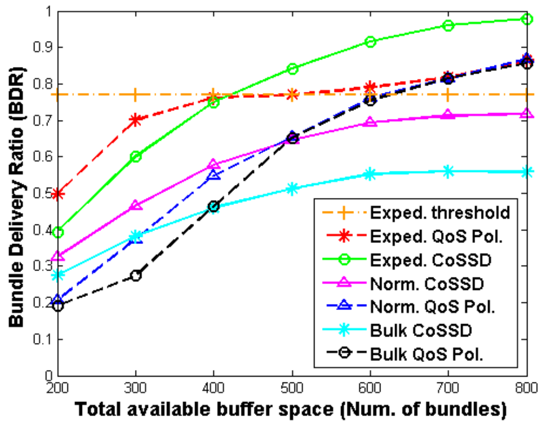

Figure 4.4: QoS Policy vs CoSSD

the Bulk class[5]. As a result, we expect from our policy to first reserve enough buffer resources to reach the performance target of the Expedited class and leave the remaining resources to the other two classes. This behavior is shown in Fig. 4.3 - 4.5, where we also compare our policy with the other two, for the case of the BDR optimization problem. To evaluate the performance of the three policies, each set of obtained BDR results, can be compared to the approximate desired performance (Table 4.3), when the buffer resources are distributed based on this logic. Obtaining such an approximate performance is straightforward, if we consider the expected number of copies per bundle required to reach a desired delivery probability and then map this to the total amount of required buffer space per QoS class. Thus, when the buffer availability is too low (i.e., 200 total spaces), corresponding to the infeasible domain, we expect not to have enough resources to satisfy even the Expedited class (i.e., $BDR = 0.77$). The other two classes should not get more resources, on average, than the ones corresponding to the minimum of a single copy per bundle throughout its lifetime (based on the constraint of Eq. (4.5)). This can be verified from Fig. 4.3 where the obtained BDR results per class comply with the expected ones. As the buffer availability increases (i.e., 300 total spaces), the first to converge to the required QoS is the Expedited class. Then, as we move towards the feasible domain (i.e., 300 - 400 total spaces), the other two classes gradually improve their performance, with the Normal class achieving steadily higher performance than the bulk one.

Inside the feasible domain (i.e., 400 - 800 total spaces) the additional resources are used to improve the performance of the lower classes and, as a result, optimize the overall network performance. This is depicted in the region 400 - 600 spaces of the figure, where the Normal and the Bulk class gradually converge to the performance of the Expedited class. Beyond that point, all of the classes achieve the same performance and exploit the complementary buffer spaces in order to further increase their BDR. We should highlight the fact that, overall, it is not until the point where the two lower classes reach the performance threshold of the Expedited class (i.e., 600 - 700 total spaces), that the BDR of the latter is increased, which is the intended behavior that leads to optimal resources distribution.

[5]As described in section 4.2.4, defining a QoS constraint for the lowest class in not necessary for our policy. Thus, it is specified in table 4.2, only for scenarios where we compare our scheme with others (i.e. BDR optimization problem)

Buffer spaces	BDR Exped.	BDR Normal	BDR Bulk
200	0.54-0.59	0.25	0.25
300	0.71 - 0.77	0.33 - 0.36	0.25
400	**0.71 - 0.77**	**0.54 - 0.59**	**0.41 - 0.45**
500	0.71 - 0.77	0.54 - 0.59	0.54 - 0.59
600	0.71 - 0.77	0.71 - 0.77	0.71 - 0.77
700	0.74 - 0.80	0.74 - 0.80	0.74 - 0.80
800	0.77 - 0.83	0.77 - 0.83	0.77 - 0.83

Table 4.3: Approximate desired performance when varying the total available buffer space

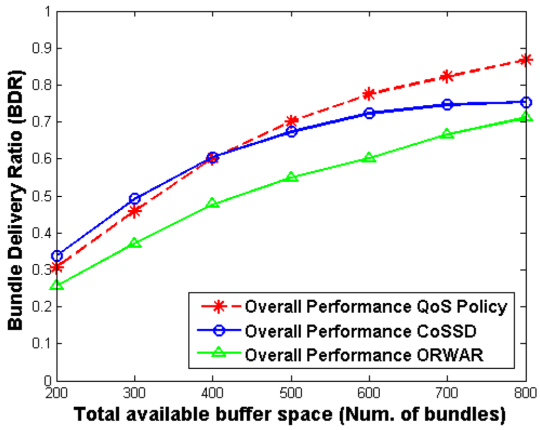

Figure 4.5: Overall Policies comparison

Regarding the comparison with ORWAR, the protocol description [79] does not specify a precise way for selecting the spraying factors per class. Since, for our simulations, all bundles are of the same length, the scheduling/dropping policy is based purely on each bundle's QoS class, by always favoring the higher class bundles over the lower class ones. Thus, to compare ORWAR with our policy, we select the replication factors per class, based on the restrictions imposed by the total buffer availability in the network and by keeping a fixed ratio among replication factors attributed to distinct QoS classes. Particularly, given a total number of available resources, n_{all}, half of the resources $\lceil \frac{n_{all}}{2} \rceil$ are distributed to Expedited bundles and, among the remaining $n_{rem} = \lfloor \frac{n_{all}}{2} \rfloor$, $\lceil \frac{2n_{rem}}{3} \rceil$ are distributed to normal class bundles and $\lfloor \frac{n_{rem}}{3} \rfloor$ to bulk class bundles [6].

Based on Fig. 4.3, it is clear that our scheme outperforms ORWAR. For low buffer values (i.e., < 500 buffer spaces), all three classes achieve higher BDR with our scheme. ORWAR fails to capture even the required performance of the Expedited class, even when the resources are adequate to do so (table 4.3). For higher buffer availabilities, ORWAR's expedited class reaches to higher BDR than the required threshold. However, this is not desired based on the previous discussion, as it comes at the cost of the other two classes, whose performance is much lower than it could be. The superiority of our scheme is also captured by the overall network performance

[6] We note that in the copies assignment rule, we also impose a minimum of 1 bundle per class, for all QoS classes.

(Fig. 4.5, considering all classes), which is up to 20% higher with our policy, comparing to ORWAR. [7]

As described in section 4.1, the derived utility function in CoSSD is based on a heuristic approach to extend [16] for the support of multiple QoS classes. Particularly, it has the following form:

$$(K - k_i) + \alpha \cdot \left(1 - \frac{m_i(T_i)}{N-1}\right) \cdot \lambda R_i exp(-\lambda n_i(T_i) R_i), \tag{4.16}$$

where k_i is the QoS class of bundle i (lower values for higher classes), K is the total number of distinct QoS classes and α is a control parameter. To compare the performance of CoSSD with our policy, we set α equal to the value for which CoSSD achieves the intended per class BDR, on average (based on table 4.3), for total buffer size equal to 400 (border of the feasible region).

In Fig. 4.4, the results of the comparison with CoSSD policy are shown. Inside the infeasible region (< 400) the lower classes, as well as the overall performance (Fig. 4.5), are improved comparing to our policy. However, this comes at the cost of significant performance degradation for the Expedited class, which does not manage to reach its required performance threshold for the first values of Buffer sizes (< 400). This is obviously contrary to the intended behavior, which dictates that our primal goal is to reach the desired performance for the expedited class. The relative behavior between the two compared policies changes inside the feasible region (> 400). The Expedited class's BDR for CoSSD increases beyond its desired QoS threshold, without the lower classes having reached this threshold. As highlighted for the comparison with ORWAR, this is opposite to the optimal behavior. The consequence is that our policy outperforms CoSSD both in terms of lower classes, as well as overall network performance in this buffer availability region.

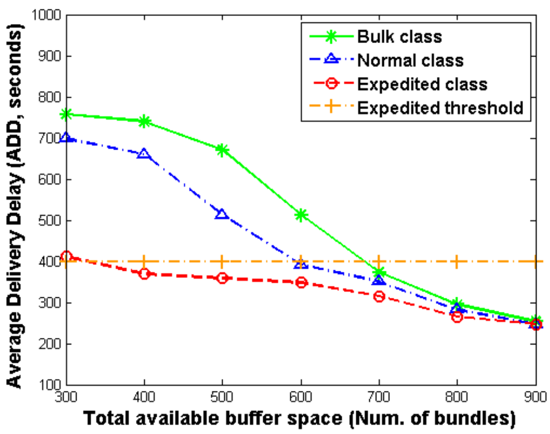

Figure 4.6: ADD Optimization problem, QoS policy

In Fig. 4.6, the performance of our scheme is evaluated with respect to the ADD optimization problem. Similarly, to the case of BDR optimization, we can verify the optimal intended per class performance here. Thus, it is shown that the Expedited class manages to stabilize its performance around the desired threshold (400 sec.), even for low buffer availability. As the

[7] We should note that other static rules for assigning copies per QoS class could have also been used. However, we claim that any such static rule would suffer from similar shortcomings, comparing to our policy and the required performance.

amount of available resources increases, the first to start improving its performance is the Normal class (300 - 500 buffer spaces), while the Expedited class remains rather stable. Then, for higher buffer availability ($\geq$ 500 spaces), it is also feasible to reach the desired performance of the Normal class (500 sec.). Thus, the additional resources are used to steadily improve the performance of the bulk class (500 - 700 spaces). Beyond that point, all three classes achieve the same ADD performance, exploiting optimally the capability offered by the resources availability.

4.4 Conclusions

In this chapter, we proposed a dynamic distributed prioritization scheme with the aim of preventing starvation of lower priority applications, while ensuring that the standards of higher priority applications are met. As discussed in sections 4.2.2-4.2.3, our policy manages to maximize the overall delivery ratio (thus, the average delivery rate), or minimize the average delivery delay metrics, among feasible solutions (i.e., solutions where the constraints of each QoS class are met). We verified the optimality of our approach through simulations based on synthetic mobility traces and we compared it with other QoS proritization approaches, showing its superiority.

As a next step, in chapter 5 we make the necessary extensions in our scheme, on top of the exponential model of pairwise inter-meeting time distributions, in order to ensure its intended performance in real life mobility conditions. Towards the same objective, in appendix A we follow another strategy, by modeling the respective distributions through the generalized pareto (power-law family) function and evaluate the corresponding performance of our scheme. As discussed in chapter 1, this approach is motivated by the strong power-law components that have been observed in the inter-meeting time patterns, based on the analysis of real mobility traces.

Chapter 5

QoS prioritization improvements for heterogeneous contact networks

5.1 Introduction

A prerequisite for the fine operation of the QoS prioritization algorithm presented in chapter 4 is to be able to make accurate predictions regarding the delivery performance, on a per message basis. Such predictions can assist optimal data prioritization and resources distribution, by dynamically keeping track of whether a message's expected metric (i.e., delivery probability or delay) is above or below its class's required threshold. In this context, capturing the pairwise inter-contact time statistics adequately can increase accuracy. In chapter 4, we consider that the pair-wise inter-contact times, being independent from each other, can be modeled through the exponential distribution, based simply on a mean pair-wise contact rate parameter $\tilde{\lambda}$, which is a characteristic of each mobility scenario. This is an assumption followed a lot in opportunistic DTN routing (e.g. [15], [16]) and works adequately for homogeneous mobility scenarios: i.e., scenarios where all pairs of nodes meet with rather the same rate (e.g. Random Waypoint, Random Walk models). However, there are multiple studies based on real mobility traces [24], [25], [26] highlighting the wide heterogeneity that can be observed in the contact rates between distinct node pairs. In such scenarios, the aforementioned model might be inadequate and lead to mis-predictions regarding the corresponding delivery metric. To this end, Serpmezis et al. [89] perform an analysis on the spreading delay of epidemic routing schemes in heterogeneous contact networks and derive closed form approximations for predicting its expected value; these approximations are still based on the exponentiality assumption but they use both first and second moments of the contact rates. In our work, we follow a similar approach to derive more accurate predictions, as well as worst-case bounds, for both metrics of interest in our optimization problems (i.e., expected delivery probability and delay). As we show through extensive simulations with real traces in section 5.3, the benefits in the performance of our scheme are significant for both optimization problems described in chapter 4.

We also propose an alternative approach to our basic scheme, accounting for scenarios where the message related information which is required for our policy (e.g. per message number of copies that exist in the network at a given time) cannot be available. This approach is based on Spray-and-Wait (SnW), instead of epidemic routing. Similarly to ORWAR, it assigns different spraying factors (number of copies) per QoS class. However, it aims at exploiting our accurate delivery estimations framework, so that the replication factors are appropriately selected to be aligned with the desired optimal performance. The results of the comparison of this approach with our basic scheme are presented in section 5.3.3.

The rest of this chapter is organized as follows. In sections 5.2.1-5.2.4, we provide the theoretical analysis to account for heterogeneous and sparse contact networks and derive the respective closed form expressions for the delivery rate and delivery delay optimization problems. In section 5.3 we evaluate the performance of our distinct approximation methods by comparing them with each other (section 5.3.2) and with other policies (section 5.3.3). In section 5.4, we conclude the current chapter.

5.2 Considering heterogeneous contact networks

As highlighted earlier, a large majority of real mobility scenarios are characterized by heterogeneous pairwise contacts (i.e., some node pairs encounter more or less frequently than others). Furthermore, some pairs of nodes may never encounter each other. If we treat such scenarios as if all the pairwise inter-meeting times follow the same distribution (with rate $\tilde{\lambda}$), we might be lead to significant prediction errors that will degrade the performance of our scheme. Our analysis so far has been based on the assumption of homogeneous pairwise contacts. In the current section, we remove this assumption and make the necessary adaptations in our schemes, in order to extend its suitability for heterogeneous contact networks. We first consider the adaptations made for the optimization of the delivery rate and then the ones for the delivery delay metric. We note that, apart from the extensions which will be described regarding our mobility model, the rest of our system model is the same as the one described in chapter 4. Similarly, we use the notation introduced in chapter 4 (table 4.1).

5.2.1 QoS optimization of delivery rate for heterogeneous contact networks

We still consider that the inter-meeting times between individual pairs are exponentially distributed; however, we now consider that the meeting rate of each individual pair is a random variable λ drawn from a probability distribution $f(\lambda)$, which is a characteristic of a mobility trace. Based on this, the probability of a bundle with n_i copies not being delivered, assuming that it has not been delivered yet, can be expressed as follows:

$$\begin{aligned} & P(bundle\ i\ not\ delivered/\ not\ delivered\ yet) = \\ & E_{\lambda}\left[\prod_{j=1}^{n_i(T_i)} exp(-\lambda_j \cdot R_i)\right] = \\ & \prod_{j=1}^{n_i(T_i)} \int_0^{\infty} exp(-\lambda_j R_i) \cdot f(\lambda_j) d\lambda_j \end{aligned} \tag{5.1}$$

Assuming that we do not know the exact distribution $f(\lambda)$, it is not possible to calculate the above probability. However, we can approximate it, based on the knowledge of the first moments of λ.

Our approach is similar to the one described in [89]. Specifically, if we consider Eq. (5.1) for $n_i(T_i) = 1$, we observe that it is the expectation of a function of a random variable λ

(i.e., $g(\lambda) = exp(-\lambda R_i)$), and thus it can be approximated through the Delta method [90], [91]. Based on the Delta method, the expectation of a function of a random variable (i.e., $E[g(\lambda)] = E[exp(-\lambda R_i)]$) can be approximated through the Taylor expansion of the function and the first moments of the variable.

Thus, we first express $g(\lambda)$ as a Taylor series expansion of its first h terms, centered at $\tilde{\lambda}$:

$$g(\lambda) = \sum_{l=0}^{h} \frac{g^{(l)}(\tilde{\lambda})}{l!} \cdot (\lambda - \tilde{\lambda})^l \tag{5.2}$$

We approximate $g(\lambda)$ by considering the first three terms of the Taylor series (i.e., $h = 2$), corresponding to the first two moments of the random variable λ. We consider that the knowledge of these two moments is a realistic assumption. Then, the approximation on $E[g(\lambda)]$ can be derived after taking the expectation of Eq. (5.2), as follows:

$$\begin{aligned} E[g(\lambda)] &= \sum_{l=0}^{2} \frac{g^{(l)}(\tilde{\lambda})}{l!} \cdot E[(\lambda - \tilde{\lambda})^l] \\ &= exp(-\tilde{\lambda} \cdot R_i) \cdot \left(1 + \frac{R_i^2 Var(\lambda)}{2}\right) \end{aligned} \tag{5.3}$$

where $Var(\lambda)$ is the variance of the meeting rates. The above expression is the probability of one of bundle i's copies not being delivered, given that it hasn't been delivered yet. Notice that, if we consider the first two instead of three terms of the Taylor series (i.e., h=1), the expression becomes the one used in section 4.2.2. To this end, we will refer from now on to the current approximation as **second order** and to the one of section 4.2.2 as **first order**, with respect to the utilized moments of variable λ. Note also that, given the convexity of $g(.)$ and, as the first order approximation is a pure function of $\tilde{\lambda}$, it is actually a lower bound on the expected probability of non-delivery, based on Jensen's inequality (i.e., $g(\tilde{\lambda}) \leq E[g(\lambda)]$). Intuitively, this means that the first order approximation is expected to give the most "optimistic" predictions, with respect to the delivery probability.

We can now express the unconditional probability of one of bundle i's n_i copies to be delivered, based on the second order approximation, as follows:

$$\begin{aligned} &E_\lambda[P_i^{(k)}(T_i)] = 1 - \left(1 - \frac{m_i}{N-1}\right) E[g(\lambda)]^{n_i} = \\ &1 - \left(1 - \frac{m_i}{N-1}\right) exp\Big(-\tilde{\lambda} n_i R_i\Big) \left[1 + \frac{R_i^2 Var(\lambda)}{2}\right]^{n_i} \end{aligned} \tag{5.4}$$

It can be shown that the above function is concave on n_i. This allows us to redefine the objective (Eq.(4.1)) and the constraint (Eq. (4.2)) functions of the optimization problem in section 4.2.2 by substituting $P_i^{(k)}(T_i)$ with $E_\lambda[P_i^{(k)}(T_i)]$ of Eq. (5.4). We can also derive the new unconstrained utilities, as follows:

$$\begin{aligned} &U_i(DR) = \frac{\partial E_\lambda[P_i^{(k)}(T_i)]}{\partial n_i} = \\ &\left(1 - \frac{m_i}{N-1}\right) \cdot exp\Big(-\tilde{\lambda} n_i R_i\Big) \Big(\tilde{\lambda} R_i - lnA\Big) \cdot A^{n_i} \end{aligned} \tag{5.5}$$

where $A = 1 + \frac{R_i^2 \cdot Var(\lambda)}{2}$. Having the new expressions of delivery probability (Eq.(5.4)) and per bundle utility (Eq. (5.5)) in hand, we can apply them in our QoS prioritization algorithm, in the same manner we did for the homogeneous case (Eq. (4.9)).

5.2.2 QoS optimization of delivery delay for heterogeneous contact networks

Considering the probability distribution of the pair-wise meeting rates, $f(\lambda)$, the expected delivery delay, $E_\lambda\left[D_i^{(k)}(T_i)\right]$, for heterogeneous contact networks can be expressed as follows:

$$\left(1 - \frac{m_i}{N-1}\right) \cdot \left(T_i^{(k)} + \int_0^\infty \frac{1}{n_i \lambda_j} \cdot f(\lambda_j) d\lambda_j\right) \tag{5.6}$$

Similarly to the case of delivery probability, the above expression can be approximated through the Taylor series expansion and the first moments of λ, as follows:

$$\left(1 - \frac{m_i}{N-1}\right) \cdot \left(T_i^{(k)} + \frac{1}{n_i \tilde{\lambda}}\left[1 + \frac{Var(\lambda)}{\tilde{\lambda}^2}\right]\right) \tag{5.7}$$

Then, the corresponding unconstrained utilities are derived by differentiating with respect to n_i:

$$\begin{gathered} U_i(DD) = -\frac{\partial E_\lambda\left[D_i^{(k)}(T_i)\right]}{\partial n_i} = \\ \left(1 - \frac{m_i}{N-1}\right) \cdot \frac{1}{n_i^2 \tilde{\lambda}}\left[1 + \frac{Var(\lambda)}{\tilde{\lambda}^2}\right] \end{gathered} \tag{5.8}$$

Thus, we can now substitute expressions 4.12 and 4.13 with 5.7 and 5.8 respectively, in the optimization problem defined in section 4.2.3.

5.2.3 Bounds on the expected performance

Although second order approximations are supposed to predict accurately enough the performance in terms of the metric of interest, it would be useful to know and exploit some bounds with respect to the expected performance. As already described in section 5.2.1, first order approximations can give us best case estimates (i.e., upper bounds for delivery probability, lower bounds for delivery delay). In the framework of our policy, though, it would be much more useful to derive bounds describing the worst case estimates. Indeed, such estimates would indicate that more resources are required to capture a given QoS threshold, thus ensuring the QoS constraints with higher consistency. Starting from convex functions of the random variable λ for both optimization problems, we can use the upper bounds derived in [92], based on the Edmundson-Madansky inequality [93] and the first h moments of λ:

$$EM_h(\lambda) = \sum_{i=0}^{h} \binom{h}{i} \frac{E[(\lambda - \alpha)^i (b - \lambda)^{h-i}]}{(b-a)^h} f\left(a + \frac{i}{h}(b-a)\right) \tag{5.9}$$

where α and b correspond to the minimum and maximum values of λ, respectively, and $f(.)$ is our convex function. It generally holds that $E[f(\lambda)] \leq EM_h(\lambda) \leq EM_{h-1}(\lambda)$. Similarly to the case of the second order approximation, we consider the second order bound (i.e., h=2).

Thus, by setting either $f = exp(-\lambda R_i(T_i))$, or $f = \frac{1}{\lambda}$, we can derive upper bounds on a single copy's expected probability of non-delivery, $EM_2^{DR}(\lambda)$, or delivery delay, $EM_2^{DD}(\lambda)$, respectively. These bounds are based only on the knowledge of the first two moments of λ and its min and max values. Then, the lower and upper bounds for the expected delivery probability and delay respectively, can be expressed as:

$$E_\lambda[P_i^{(k)}(T_i)] \geq 1 - \left(1 - \frac{m_i}{N-1}\right) \cdot [EM_2^{DR}(\lambda)]^{n_i} \tag{5.10}$$

$$E_\lambda\left[D_i^{(k)}(T_i)\right] \leq \left(1 - \frac{m_i}{N-1}\right) \cdot \left(T_i^{(k)} + \frac{EM_2^{DD}(\lambda)}{n_i}\right) \tag{5.11}$$

5.2.4 Considering sparse contact networks

Based on the previous analysis, our policy requires estimates on the first and second moments of the pairwise meeting rates, $\tilde{\lambda}$ and σ_λ^2, which characterize each mobility trace. These estimates are produced through the estimates on the individual meeting rates $\lambda_{i,j}$ which characterize each pair of nodes $< i, j >$. However, $\tilde{\lambda}$ and σ_λ^2 are extracted considering only the pairs of nodes that encounter at least once during the duration of the trace.

Nonetheless, in real traces there is usually a large portion of node pairs that never encounter (i.e., $\lambda_{i,j} = 0$). Let's consider a contact graph where each node is represented as a vertice and an edge between two vertices exists only if the corresponding nodes encounter at least once during the trace. Then, the lack of edges between some pairs of nodes would lead to a sparse contact graph. As described in [89], such a graph can be modeled as a Poisson contact graph where, for each pair of nodes $< i, j >$, there is probability p_s that they will be meeting with rate $\lambda_{i,j}$ and probability $1 - p_s$ that they will never meet. Based on this model, it is shown in that the contact rate moments should be altered as follows:

$$\begin{aligned} \tilde{\lambda}_{p_s} &= p_s \cdot \tilde{\lambda} \\ \sigma^2_{\lambda_{p_s}} &= p_s \cdot (\sigma^2_\lambda + \tilde{\lambda}^2 \cdot (1 - p_s)) \end{aligned} \tag{5.12}$$

where p_s is a density characteristic of each trace, which can be approximated as the ratio of the total number of pairs that encounter throughout the trace, over the total number of distinct pairs that exist in the network $\binom{N}{2}$.

5.2.5 Discussion

Throughout our analysis, we have considered QoS requirements applied with respect to the "expected" values for the metric of interest. However, our framework could be appropriately adopted to support more generalized QoS requirements. Thus, an alternative policy can consider tighter constraints to ensure the requirements satisfaction. For example, one could impose that the delivery delay per bundle should be lower than its class's threshold, with a probability larger than 90 %. Of course there is a tradeoff there. The more conservative a policy is, the fewer the instances when it is actually missing the threshold, but the more the cases when it is wasting too many resources, just to ensure constraints satisfaction. The generalization of QoS requirements is beyond the scope of this work, but, at the end, the design of a policy boils down to which is the primal goal: meet the constraints at all costs, or optimize the performance, while meeting the constraints, on average, and not wasting resources.

Another interesting problem, which we are planning to address in future work, could consider both traffic sensitive to delivery ratio and traffic sensitive to delay, concurrently. This would imply imposing both the respective types of constraints in our system. The solution to such a problem would be of practical use in many scenarios where applications having either type of requirements are launched in the same time.

5.3 Performance Evaluation

In the following, we compare the performance of our new approximation methods with the first order approximations of chapter 4, as well as with the other prioritization policies we were compared to, based on real mobility traces this time. Particularly, we consider: 1. The Cabspotting trace which is based on tracking the movement of 536 taxis in San Fransisco [94], 2. The Infocom trace [95] originating from Bluetooth sightings of 98 nodes during 4 days in the Infocom 2006 conference. The configuration parameters for the simulations based on the two different mobility scenarios are summarized in table 5.1.

	Cabspotting trace	*Infocom trace*
Number of Nodes (N)	536	98
Total simulation time (min.)	440	335
Mean pairwise meeting rate ($\tilde{\lambda}$, min^{-1})	$4.1 \cdot 10^{-3}$	$2.3 \cdot 10^{-2}$
Pairwise meeting rates variance (σ^2_λ, min^{-2})	$8.9 \cdot 10^{-6}$	$4.6 \cdot 10^{-4}$
Density coefficient (p_s)	47%	68%
Bundle TTL (BDR Optimization problem, sec.)	6000	3000
Bundle TTL (ADD Optimization problem, sec.)	15000	4000
Mean rate of contact window r_d (% of unconstrained rate)	0.5	0.5
Expedited desired QoS (BDR opt.)	0.74	0.77
Normal desired QoS (BDR opt.)	0.35	-
Expedited desired QoS (ADD opt, sec.)	3900	560

Table 5.1: Simulation Parameters based on real mobility traces

5.3.1 Evaluation Setup

For both traces, we focus on time windows where the total number of meetings per hour does not change significantly. This time, we consider the competition among two distinct QoS classes (i.e., expedited and normal class). The performance is evaluated for both delivery rate and delivery delay optimization problems, as we vary the buffer spaces availability. As observed from the analysis of the Cabspotting and Infocom traces, the respective networks are not fully mixed (i.e., $p_s < 1$, table 4.2), within the duration of the time windows that we investigate. Based on

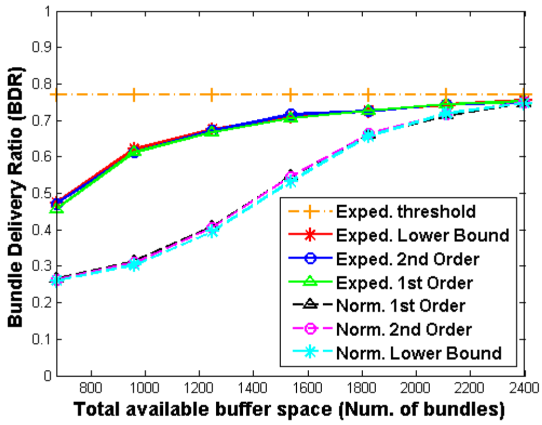

Figure 5.1: Infocom: BDR Optimization, Approximations comparison

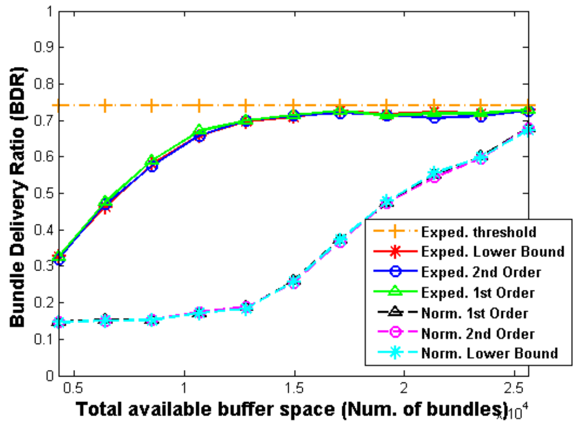

Figure 5.2: Cabspotting: BDR Optimization, Approximations comparison

this observation, the mean and the variance of meeting rates for each trace are extracted based on Eq. (5.12).

5.3.2 Comparison among approximations

In Fig. 5.1 - 5.10, we evaluate the performance of our policy when its implementation is based on the three different approximation approaches (i.e., first order, second order and bound approximations). In terms of the **BDR optimization problem** (Fig. 5.1 and 5.2), it can be verified that all the three approaches manage to achieve the intended optimal performance, as for the case of the synthetic traces. Particularly, for both traces, the Expedited class stabilizes its BDR around the desired value (i.e., 0.77 for Infocom, 0.74 for Cabspotting) and, as the buffer resources further increase, the normal class steadily improves its performance up to the point where the two curves converge. This behavior is captured ideally in the case of the Cabspotting trace (Fig. 5.2). In the case of the Infocom trace (Fig. 5.1), it can be noticed that, for low buffer availability, some of the Normal class bundles start getting delivered without the Expedited class having totally stabilized at its QoS threshold. This can be justified by the source copy restriction that we impose (Eq.(4.5)) even for Normal class bundles. Given the smaller size of the network comparing to the Cabspotting trace, the relative impact of a single copy on the delivery performance is greater in the Infocom trace. Thus, this restriction in combination with the limited resources availability, do not permit the Expedited class BDR to perfectly converge to the required value, before the Normal class starts increasing its own BDR.

The fact that no difference is observed in the performance of the three policies can be explained by inspecting Fig. 5.3 and 5.5. There, the average number of copies per bundle per class is drawn, throughout the bundle's lifetime for some fixed amount of buffer availability. It can be observed that, with the second order and lower bound approximations, more copies are distributed to Expedited class bundles at the beginning of their lifetime, as opposed to the respective copies distribution with the first order approximation. This comes as a result of the second order and lower bound approximations making more "conservative" predictions, with respect to the bundle delivery probabilities. Consequently, they indicate that more copies are

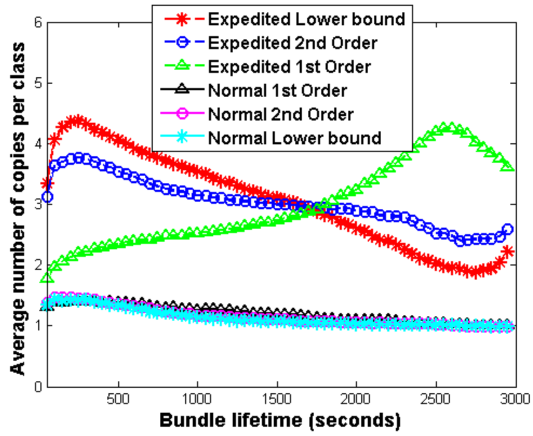

Figure 5.3: Infocom: Copies over time, Approximations comparison (Buff= 10)

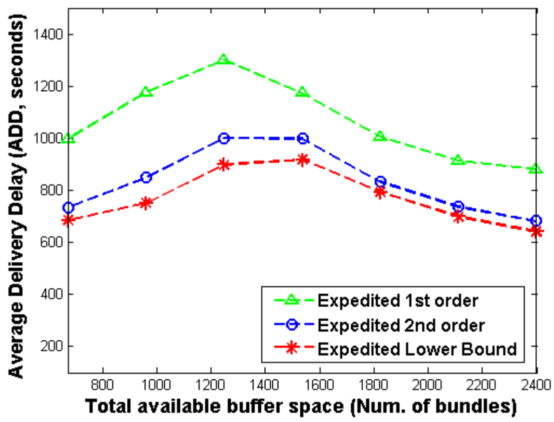

Figure 5.4: Infocom: BDR Optimization, ADD metric Approximations comparison

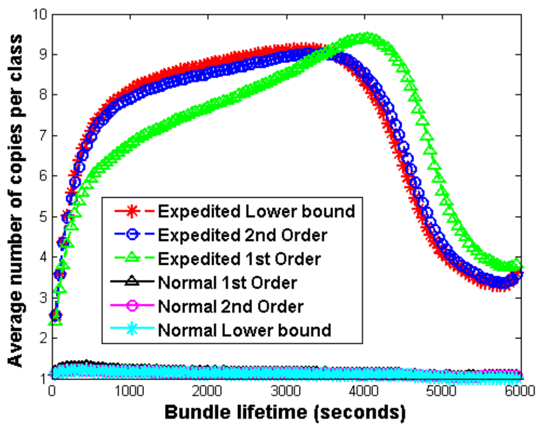

Figure 5.5: Cabspotting: Copies over time, Approx. comparison (Buff= 20)

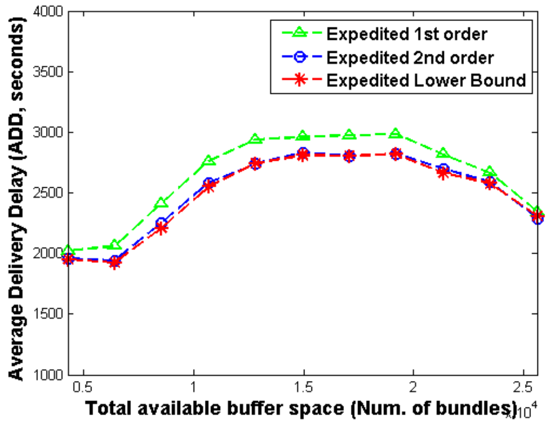

Figure 5.6: Cabspotting: BDR Optimization, ADD metric Approximations comparison

required to reach the desired Expedited class performance. However, as described in section 4.2.2, the predictions are also based on dynamically monitoring the percentage of "seen" nodes $\left(\frac{m_i}{(N-1)}\text{ in Eq. (4.7) }\right)$. When this percentage is low, with respect to the remaining TTL of the expedited class bundles and the desired QoS threshold, our policy can compensate the possible "over-optimistic" initial predictions, by distributing more copies to them, as they approach at the end of their lifetime. This behavior is more obvious in the case of the first order approximation for the Infocom trace (Fig. 5.3); It can be explained by the trace's higher heterogeneity with respect to pairwise contact rates, which makes it harder to make accurate predictions based on the first order approximation.

Although the delivery probability mispredictions of the first order approximation can be compensated in the manner we described, the same doesn't occur when we examine the ADD performance for the Expedited class (Fig. 5.4 and 5.6, yet still in the context of the BDR optimization problem). There, it is evident that the distribution of more copies at the beginning

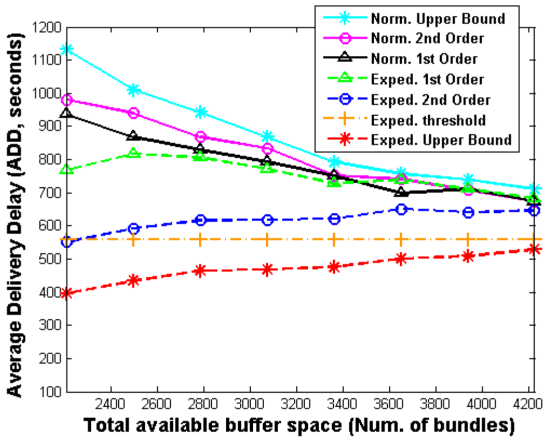

Figure 5.7: Infocom: ADD Optimization, Approx. comparison

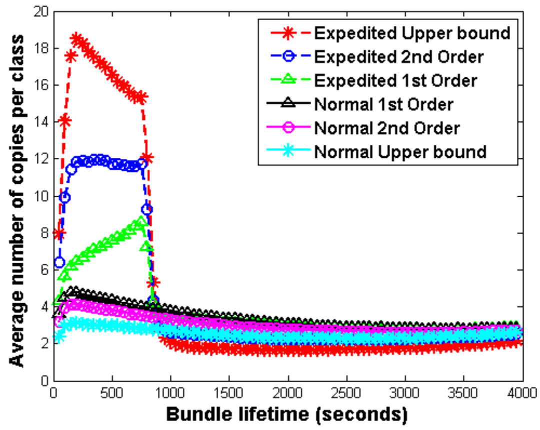

Figure 5.8: Infocom: Copies over time, Approx. comparison (Buff= 23)

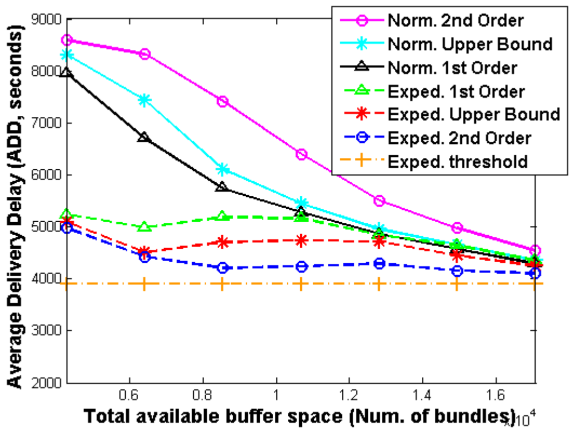

Figure 5.9: Cabspotting: ADD Optimization, Approx. comparison

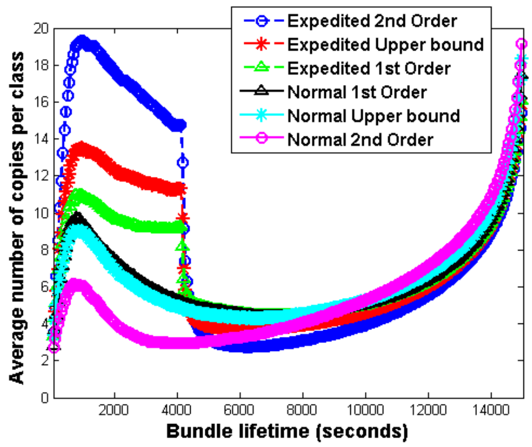

Figure 5.10: Cabspotting: Copies over time, Approx. comparison (Buff= 20)

of the bundles lifetime by the two more conservative approximations permits to decrease the average delivery delay, comparing to the first order approximation. Intuitively, this makes sense if we consider that, by distributing more copies at the beginning of a bundle's lifetime, it is more likely that one of them will encounter the destination sooner. Thus, the usefulness of the better approximations is evident for the BDR optimization problem, as the results in terms of the delay metric are improved, without compromising something else. At this point, we should highlight that the ADD performance is extracted from the delivered messages only. This explains why, in some cases, the increase in buffer space availability is accompanied by an increase, instead of decrease, in the ADD performance. Particularly, when the delivery ratios are lower, it is more likely that the fewer messages that get delivered do so in relatively shorter time, than when they are higher. This also explains analogous behavior in the results of Fig. 5.7 and 5.9.

Let's turn our discussion now to the evaluation with respect to the actual **ADD optimization problem** (Fig. 5.7 - 5.12). Based on our simulation framework, a bundle's required delivery

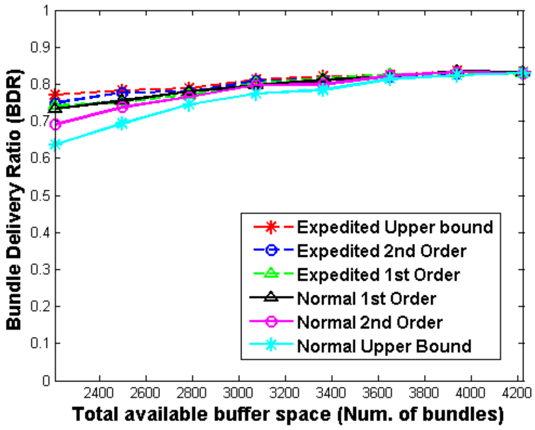

Figure 5.11: Infocom: ADD Optimization, BDR metric Approximations comparison

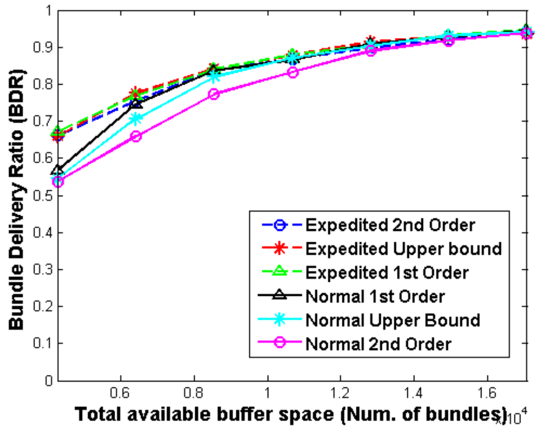

Figure 5.12: Cabspotting: ADD Optimization, BDR metric Approximations comparison

delay (560 seconds for Infocom, 3900 seconds for Cabspotting trace) is different from its TTL, which is actually much larger (i.e., 4000 sec. for Infocom, 15000 sec. for Cabspotting) to permit for delivery probability close to 100%. In Fig. 5.11 and 5.12, the BDR per class performance of the two traces is depicted, for the three different approximation approaches. Once again, the second order and upper bound-based implementations of our algorithm distribute more copies to the Expedited class bundles at the beginning of their lifetime (Fig. 5.8 and 5.10)[1]. Contrary to the BDR performance of the respective optimization problem though, this has a crucial impact on the performance of our scheme. The two more conservative approximations manage to capture the QoS requirement of the Expedited class better, for all the range of buffer values (and both traces), as opposed to the first order approximation, whose mispredictions do not allow to do so (Fig. 5.7 and 5.9). Of course, the distribution of more copies to the Expedited class, leaves less resources to the Normal class and, as a result, the latter's performance is better with the first order approximation than with the other two, for a range of buffer values. However, once again we verify the intended behavior of the algorithm: *as the buffer availability increases, the delivery delay of the normal class is constantly decreasing with the conservative approximations, while the Expedited class remains stable around its threshold*.

Regarding the comparison between the second order and (upper/lower) bound approximations, for both optimization problems, the following can be observed. For the Infocom trace, it is evident that the ADD of the Expedited class with the upper bound is constantly below the one with the second order approximation for all the evaluated scenarios (Fig. 5.4 and 5.7). As explained in section 5.2.3, this is foreseen, as the predictions with the bound approximations are expected to be more conservative than the ones of the second order approximation (Fig. 5.3 and 5.8). For the Cabspotting trace, though, the performance differences between the second order and bound approximations are smaller and, for the case of the delivery delay optimization problem, the ADD with the second order approximation is even slightly lower than the respective

[1]We note that the steep decrease in the number of copies of the expedited class for $T_i^{(k)} > D_{QoS}^{(k)}$ is dictated by our algorithm which imposes that copies of bundles which have exceeded their QoS threshold cannot be classified in the high priority group (section 4.2.3).

one with the upper bound (Fig. 5.9). Although not expected, this can be due to the distribution of meeting rates during the selected time window of the trace, which leads to more optimistic estimations of the delivery delay than the second order approximation (Fig. 5.10).

5.3.3 Comparison with other policies

In the following, we compare the performance of our policy with other policies. In Fig. 5.13, the per class performance of our policy is compared to ORWAR and CoSSD, with respect to BDR optimization. The configuration of the two other policies has been done in the manner described in chapter 4 (section 4.3.2). It is obvious that our policy outperforms ORWAR for both classes. This of course has an impact on the overall performance (Fig. 5.14), where our scheme achieves up to 20% higher results than ORWAR. Regarding the comparison with CoSSD, similarly to the synthetic simulation results (section 4.3.2), it is clear that CoSSD fails to capture the intended per class (Fig. 5.13) and overall performance (Fig. 5.14). Our policy steadily outperforms it in terms of Expedited class BDR, inside the infeasible domain (i.e., 4250-12800 buffer spaces), at the cost of the Normal class's performance; inside the feasible domain (i.e., $\geq$ 17000 buffer spaces) the picture changes, with our scheme's Normal class BDR exceeding the respective one with CoSSD, while the Expedited class remains stable around the desired threshold. Notice that, for both domains, the performance difference in terms of Normal class BDR between the two policies is much larger than the respective difference in terms of the Expedited class. This can be explained considering that, given the same amount of additional resources (bundle copies), the performance gain for bundles which are given a low number of initial copies on average (i.e., Normal class) can be much higher than the respective gain for bundles with an already high number of copies (i.e., Expedited class).

In Fig. 5.14, the overall BDR performance of our scheme is also compared to the optimal QoS unconstrained scheme of Krifa et al. [16], which considers a single priority class. It is evident that the unconstrained scheme achieves higher performance than our policy. This makes sense since the primal aim of our scheme is to satisfy the constraints of the higher classes. Thus, when the resources are limited, this has an impact on the overall network performance degradation comparing to the unconstrained case. However, when the resources are enough to allow the normal class to start converging to the performance of the expedited class (i.e., $\geq$ 15000 spaces), the overall performance of our scheme also starts to converge to the one of the single class scheme. This is another indicator of the optimal intended behavior of our policy.

In Fig. 5.15 - 5.16, the comparison of our policy with a more "optimized" version of the ORWAR protocol is depicted. Similarly to ORWAR, we use SnW and assign different replication factors per QoS class, proportional to their importance. However, this assignment is not done based on a static rule, like the one we used for ORWAR. Instead, it is done based on the logic specified in section 4.3.2. Thus, resources permitting, n_{exp} copies are always given to Expedited class bundles, to ensure its QoS requirement. If there exist remaining resources, $0 < n_{rem} < n_{exp}$, after this assignment, they consist the replication factor for Normal class bundles. Finally, if $n_{rem} > n_{exp}$, both classes are given equal initial number of copies, towards the target of overall performance maximization. For the BDR optimization problem, similarly to ORWAR, the higher class bundles are given absolute priority over the lower class ones. For the ADD optimization problem, though, to prevent starvation of the normal class, expedited class bundles are given absolute priority only while it holds that $T_i^{(k)} < D_{QoS}^{(k)}$. For higher $T_i^{(k)}$ they get the same priority as normal class. Finally, for both problems, bundles of the same class are prioritized in

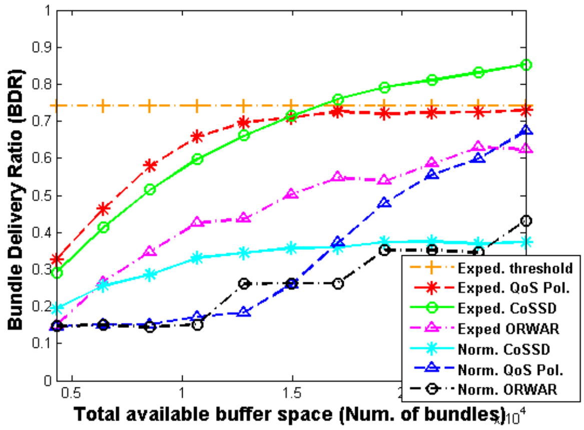

Figure 5.13: QoS policy vs ORWAR and CoSSD (Cabspotting)

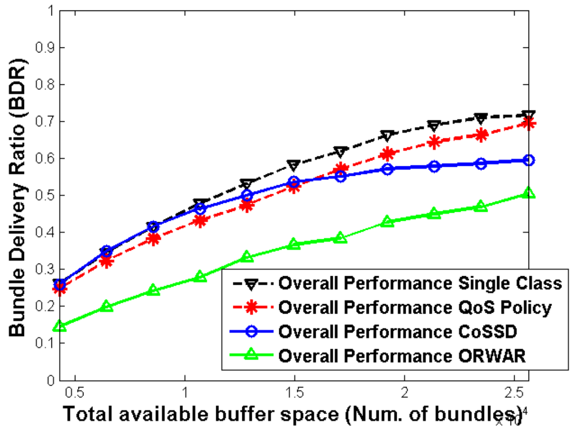

Figure 5.14: Overall policies BDR comparison (Cabspotting)

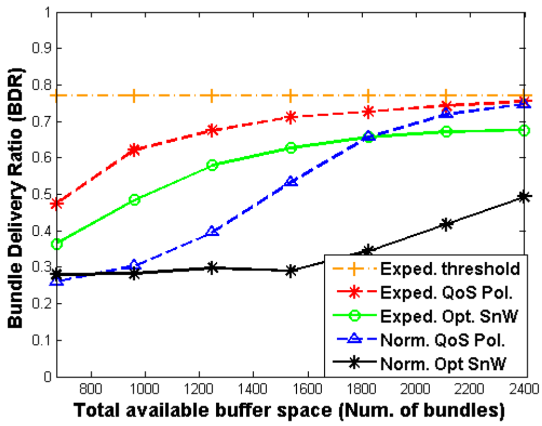

Figure 5.15: QoS Policy vs Opt. SnW (BDR, Infocom)

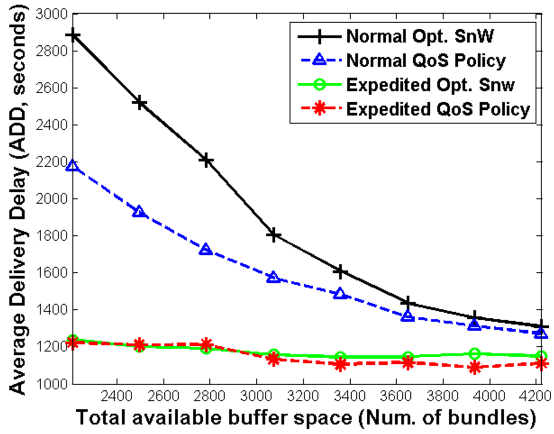

Figure 5.16: QoS Policy vs Opt. SnW (Normalized ADD, Infocom)

descending order of their remaining TTL [2].

From Fig. 5.15, it is evident that the aforementioned approach performs worse than our basic policy. Contrary to our scheme, it cannot dynamically adjust the number of copies each bundle is getting, in order to cope with possible initial delivery probability mis-predictions. Thus, even if, comparing to our prediction, a lower number of copies is required [3], on average, to reach the desired threshold, the subset of bundles which do not get delivered contribute to the worse performance comparing to our policy. As a result, the BDR of the Expedited class is constantly below the required threshold and the respective curve with our basic policy (5% - 15%). Regarding the performance of the Normal class, we notice that, although below our basic policy as well, its "starvation" is prevented, for increasing availability of resources (i.e., $>$ 1600

[2] As it is shown in [16] this simple "drop oldest bundle" (or "schedule youngest") can be a good approximation of the optimal unconstrained utilities, when the congestion regimes in the network are low.

[3] To determine the required number of copies the delivery predictions based on the lower bound approximation were used (most conservative approximation).

total spaces), in a more efficient manner than with the previous configuration of ORWAR (Fig. 5.13). This performance is indicative of a more desired behavior with respect to our optimization problem.

In terms of the ADD optimization problem, due to the large difference which was observed in terms of BDR per QoS class (up to 23% higher Normal class BDR with the basic policy), a direct comparison in terms of delivered bundles ADD wouldn't be fair. Thus, we compared the two schemes with respect to a normalized ADD metric which considers the non-delivered bundles as well. Particularly, for a BDR value x and an ADD value y (of delivered bundles), the normalized ADD is computed as: $x \cdot y + (1 - x) \cdot TTL$. Although the performance in terms of the Expedited class is practically the same, it is obvious that the basic QoS policy outperforms the other scheme, in terms of the Normal class and, thus, the overall network performance.

Overall, we claim that the optimized SnW scheme can consist a decent alternative to our QoS policy, considering its reduced overhead comparing to the latter. Indeed this "myopic" approach doesn't require any statistics collection for the prediction of n_i and m_i, as the QoS policy does. Instead, the resources distribution is purely based on the initial predictions, with respect to the required number of copies per class.

5.4 Conclusions

In this chapter, we suggested efficient extensions to the QoS prioritization policies introduced in chapter 4, in order to be able to guarantee the intended performance in real life mobility conditions, characterized by heterogeneous and sparse contacts. Moreover, we suggested an alternative scheme based on SnW. Although performing worse than our basic scheme, it can be appropriate for cases where we need to balance the trade-off between ease of use and high performance guarantees. We verified the optimality of our approach through simulations based on real mobility traces and we compared it with other QoS proritization approaches, to validate its superiority.

Our approach here was based on making more accurate delivery predictions by exploiting the second moment of the pairwise meeting rate random variable λ, while still modeling the inter-meeting patterns through the exponential distribution. As mentioned in chapters 1 and 4, in appendix A we also follow a different way by modeling the pairwise inter-meeting time distributions through the generalized pareto function. The performance of the two different approaches is compared through extensive simulations based on real mobility traces, considering both optimization problems of interest.

Chapter 6

Conclusions and Future Research

In this thesis, we proposed a complete QoS prioritization framework for resource constrained opportunistic DTNs. This framework is based on optimizing the scheduling and buffer management decisions, in a distributed manner, according to the per bundle dynamic delivery predictions and the QoS classes delivery requirements. To this end, the performance of our policies strongly depends on the accuracy of such predictions. In this context, we also focused on making the necessary adaptations in our QoS policies in order to account for real world mobility conditions and pairwise contact patterns. Our contributions can be summarized in the following:

- We provided an extensive classification of existing state of the art congestion control and reliability approaches for opportunistic DTNs in the literature (chapter 2). Based on this classification, we showed the dependencies of each solutions framework to the associated networking environments (e.g., mobility and nodes contact patterns, types of disruptions), the data routing approach (e.g., single copy vs multiple copy routing), the approaches objectives (e.g., congestion avoidance vs congestion management) and their generic operation principles. Moreover, through this review, we provided a reasoning for the design choices of our selected approach, in terms of QoS prioritization.

- We formulated the problems of QoS prioritization as constrained optimization problems, with the aim of maximizing the network performance for a specific metric (i.e., delivery rate maximization, or delivery delay minimization), while respecting the individual QoS requirements and storage resource constraints (chapter 4). The purpose was to optimize the resources allocation by first ensuring the satisfaction of individual QoS class constraints in the order of their importance and, then, allocate any remaining resources towards the performance maximization target which is equivalent to avoiding the starvation of lower application classes. We provided a distributed solution to this problem, based on deriving appropriate per bundle utilities which express each bundle's marginal value, with respect to the network's optimization objective function. To this end, we also provided a simple implementation of our schemes, which specifies the practical buffer sorting principles, so as to be aligned with our theoretical analysis and guarantee the intended performance. The optimality of our approach for both optimization problems was verified based on synthetic trace simulations and comparisons with other existing policies.

- We suggested extensions in our policies in order to ensure the intended delivery performance in real life mobility conditions. These extensions are related to capturing the appropriate pairwise inter-contact time statistics more adequately, comparing to the approach described in chapter 4. To this end, in chapter 5 the pairwise contact rates homogeneity assumption is removed and the second moment of the pairwise contact rate random variable is appropriately exploited, still based on the assumption of exponentially distributed inter-contact times, in order to derive more accurate delivery predictions, as well as worst case delivery performance bounds. As explained in the respective chapter, these bounds can be useful in order to ensure the desired sequence of performance targets (i.e., satisfy the QoS class delivery requirements from the highest to the lowest one) by making more "conservative" delivery predictions, with the cost of lower overall performance. The performance benefits of our extensions, comparing to the first order exponential model, were demonstrated through simulations based on real mobility traces. We also proposed an alternative policy based on using Spray-and-Wait, instead of Epidemic routing, and exploiting the aforementioned benefits of our higher order delivery approximations, to optimally determine the initial replication factors per QoS class. Although its delivery performance was worse than our basic scheme, we claim that it can be useful for cases where the collection of the statistics required for the operation of our basic approach (i.e., estimation of n_i, m_i) is not available for any kind of reason.

In appendix A, we described a different approach by modeling the inter-contact times through the generalized pareto, instead of the exponential distribution. This approach was motivated by existing analysis in the literature, highlighting the strong power-law components observed in the inter-contact times of many real mobility traces. As shown through the respective simulation results, although this approach performed better than the first order exponential for some of the evaluated scenarios, it didn't offer any significant performance benefit comparing to the higher order exponential approximations. On the contrary, it performed worse than the latter for the case of the delivery delay optimization problem.

Future Research

In the framework of our QoS prioritization schemes, we have so far considered performance maximization objectives and QoS requirements per class referring to the same type of metric (i.e., either delivery rate (ratio), or delay). As highlighted in chapter 5, an extension of practical interest for many scenarios would be one that considers both applications sensitive to delivery ratio and applications sensitive to delivery delay, launched at the same time and competing on the available resources. We claim that such a problem could also be solved based on the distributed optimization framework we have adopted, this time considering both types of constraints in the system. For example, the performance maximization objective could still be expressed with respect to one of the metrics of interest, by penalizing either the delivery ratio or delivery delay constraint violations from the respective QoS classes. A different approach could consider the joint delivery ratio and delivery delay optimization problem, by defining objective functions which would appropriately weight the contribution of each bundle, according to its corresponding metric of interest and delivery predictions.

Another direction of practical interest, towards the validation of our policies in a real implementation, would be to relax the assumption on the global knowledge of the number of replicas per bundle and evaluate their performance, when they are based on some copies estimation method, instead. As highlighted in chapter 4, such an efficient method has been proposed in [16] and tested for the case of single QoS class problems.

Furthermore, throughout our analysis we have considered unicast applications. However, we claim that in many opportunistic contact scenarios of interest, multicast applications could also benefit from a similar QoS prioritization framework. For example, going back to the military use case scenario examined in chapter 3, it can be expected that more than one nodes would be interested in getting the position updates of a specific node. Similarly, if we consider social network applications, multiple users could be interested in locally available common data contents. To this end, it would be essential to appropriately adopt our policies in order to express and ensure QoS requirements such as a required probability of delivery to all the members of a multicast group.

Another interesting direction would be to consider energy constraints, together with the existing resource and QoS constraints, in the distributed optimization framework we have adopted. This would be of practical use for scenarios where energy limitations can be present (e.g., dismounted soldiers carrying portable devices in military networks, users smart phones in social networks). Throughout our framework, we have considered scheduling and congestion management techniques running on top of epidemic routing. Although epidemic routing is very aggressive with respect to storage resources consumption, through our distributed framework we manage to counteract the negative effects of this aggressiveness by optimally allocating the available resources, according to the intended delivery performance. However, energy resources consumption from the redundant amount of transmissions, imposed by epidemic routing, cannot be counteracted. To this end, an appropriately adopted framework which can guarantee the desired QoS based performance, while respecting the energy constraints, needs to be considered. Such constraints could be expressed for example as a minimum required time (e.g., duration of a military mission) until the mobile devices start to run out of battery.

On a different aspect, throughout this thesis we have considered scheduling and congestion control approaches, based on distributed resource allocation decisions for data traffic which is already circulating in the network. We claim that in many scenarios it would also be feasible and effective to control the amount of traffic which is injected in the network from each QoS class. Thus, for example, the transmission of less time critical traffic could be suspended by the source node, in favor of more urgent traffic, when there are some form of indications for high congestion regimes in the network. In this context, data traffic admission control hasn't been extensively addressed in the DTN literature, especially considering opportunistic contact networks. To this end, it would be interesting to examine how to optimally address this problem.

Chapter 7

Résumé [Français]

7.1 Contexte des Réseaux DTN

Les capacités de communication mobile augmentent aujourd'hui en termes de connectivité et de débit des données, ainsi que de la large gamme des applications qu'elles offrent. Les utilisateurs peuvent exploiter différents types des interfaces (par exemple, 4G / LTE, WiFi, etc.) et des types des réseaux (par exemple, basée sur l'infrastructure ou pas), selon le type d'application (par exemple l'augmentation du trafic du réseau via une interface peut déclencher le déchargement du trafic de l' utilisateur via une autre interface).

Les réseaux ad-hoc mobiles (MANETs) sont une famille des réseaux caractérisés par une communication directe entre les utilisateurs mobiles, sans aucun support d'une infrastructure centralisée (par exemple, des stations de base cellulaires ou des points d'accès sans fil). Egalement, tout noeud (utilisateur) peut agir comme source des données, destination ou relais, pour transmettre un message à sa destination. Dans ce contexte, l'utilisation des MANETs est populaire pour les services de portée locale (par exemple, les réseaux ad hoc véhiculaires pour les applications de transport / sécurité, les réseaux des capteurs sans fil, les réseaux militaires / tactiques, les réseaux sociaux mobiles, etc.). En raison du mouvement des utilisateurs, la topologie de ces réseaux évolue dynamiquement et ainsi la construction des chemins de routage des données entre les paires de source et destination est difficile. Cependant, il est supposé qu'ils existent toujours des chemins de routage de bout en bout, de n'importe quelle source vers n'importe quelle destination. Néanmoins, il existe des scénarios réels pour lesquels une telle hypothèse serait irréaliste, en raison des perturbations de connectivité fréquentes et / ou longues entre les pairs des noeuds que communiquent, conduisant aux réseaux intermittents (ICN). Ces perturbations peuvent résulter des topologies des réseaux clairsemées, des obstacles de terrain, de la mobilité des noeuds ou des contraintes des ressources (bande passante par opportunité de communication, stockage, énergie).

Dans ce contexte, les Réseaux tolérantes aux petrurbations (DTNs) peuvent être considérés comme un type spécial de MANETs qui vise à fournir des services de communication dans des telles conditions stressantes. Pour survivre la connectivité intermittente, les DTNs sont basés sur le concept "store-carry-and forward": les noeuds mobiles peuvent stocker leur propre contenu, ou celui d'autres noeuds, jusqu' à ce qu' une prochaine opportunité de communication apparait, soit avec la destination (distribution de contenu) soit avec un relais auquel ils peuvent transmettre des données. Sur la base de ce principe, les applications DTN s'exécutant sur les hôtes finaux, peuvent rester transparentes des interruptions de connectivité. Cependant, le manque de connectivité bout en bout, les possibilités de communication limitées, ainsi que l'exigence pour les noeuds de stocker leurs propres données et ceux d'autres noeuds dans des environnements des ressources limitées, rend très difficile de garantir la delivrage des données avec des délais précis.

Dans ce contexte, le transport des données basé sur TCP / IP se révèle être très inefficace dans la plupart des cas. Le protocole de contrôle de la transmission (TCP) [2] et plusieurs de ses extensions (par exemple, [3], [4], [5]) sont très populaires pour des nombreuses applications d' Internet qui exigent la fiabilité (i.e. assurer la livraison des données aux destinataires prévus) et la prise en charge du contrôle de la congestion. Cependant, son fonctionnement est basé sur l'existence d'une connectivité continue de bout en bout, qui est généralement absente dans les types d'environnements des réseaux susmentionnés. De plus, d'une part, les mécanismes TCP ne peuvent pas tolérer les grands retards induits par les réseaux intermittents connectés. D'autre part, plusieurs opérations de TCP (par exemple l'établissement de connexion, la contrôle de la congestion basée sur rétroaction) constituent habituellement un très grand overhead pour les DTNs qui peuvent dégrader leur performance de manière significative, compte tenu la limitée quantité et durée des possibilités de communication.

Selon l'environnement du réseau, DTNs peuvent donner des alternatives fonctionnelles aux approches semblables au TCP, afin de fournir une communication fiable, lorsque cela est faisable, ou des approches du meilleur effort (best effort) lorsque cela n'est pas possible. Ces approches pourraient ne pas être en mesure de garantir la livraison des données, mais leur objectif est de maximiser la performance du réseau, compte tenu des conditions des contraintes susmentionnées.

7.1.1 Types de réseaux et profiles de mobilité

Les noeuds mobiles peuvent échanger des données lorsqu'ils sont à portée de communication (c'est-à-dire lorsqu'ils entrent en contact, sur la base de la terminologie DTN) et les conditions externes le permettent. Dans ce contexte, il y a deux considérablement différent types des contacts qui peuvent exister, sur la base des profiles de mobilité des noeuds:

- *Contacts programmés / prédéterminés:* les paires de noeuds se rencontrent, sur la base d'un horaire spécifique qui peut être connu. Ce type des contacts est habituellement observé dans les réseaux interplanétaires, où les parties communicantes (par exemple les satellites) se déplacent sur des orbites prédéterminées et, par conséquent, les opportunités de contact apparaissent périodiquement. Les scénarios de cas d'utilisation où les contacts sont prédéterminés peuvent également être observés dans le cadre des réseaux terrestre. Par exemple, dans [6], [7] "messages ferries" sont utilisés pour recevoir et transmettre les données des utilisateurs aux passerelles, fournissant ainsi un accés Internet à faible cout. Dans des tels scénarios, le routage des données peut être statique et même des chemins de bout en bout entre la source et la destination peuvent être exploités. Ainsi, le principal facteur de contrainte provient ici des grands retards causés par la durée entre contacts consécutifs (inter-contact times), ou du temps de propagation et des erreurs de canal dans des réseaux interplanétaires.

- *Contacts opportunistes:* les paires de noeuds se déplacent sans aucun horaire et / ou itinéraire spécifique, et alors ils entrent en contact et peuvent échanger des données pendant des occasions inattendues. Dans ce contexte, il est impossible de savoir quand le prochain contact avec un noeud spécifique aura lieu. Par conséquent, il est également habituellement impossible de construire des chemins de routage valides et à jour vers les noeuds de destination. Des types des contacts opportunistes peuvent être trouvés dans un large éventail des scénarios terrestres où les solutions DTN sont appliquées (par exemple, les réseaux sociaux mobiles, les DTN véhiculaires ou militaires).

7.1.2 Cadre des solutions DTNs et relation avec le routage des données

Sur la base de l'approche "store-carry-and-forward", les messages des données peuvent être transmis par plusieurs noeuds des relais avant d'atteindre le (s) destinataire (s) final (aux). Dans ce contexte, la fiabilité hop-by-hop (saut-à-saut) est suggérée, comme une alternative possible à la fiabilité de bout en bout. Plus précisément, un message est acquitté au saut précédent (relais) avant le passage au saut suivant. Ensuite, le nouveau relais stocke le message et il est responsable soit de sa livraison vers la destination, soit de son acheminement vers le prochain saut du chemin de routage. Sur la base de ce cadre, un chemin de livraison de bout en bout peut être divisé en plusieurs sous-chemins, où, pour chaque sous-chemin, la fiabilité, dans la perspective d'assurer la livraison des données peut être assurée. Aussi, si les horaires des opportunités de communication, parmi les noeuds DTN comprenant le chemin de bout en bout, sont plus ou moins connus, le délai de livraison des données à la destination peut également être estimé. Ainsi, cette approche peut être adéquate pour les scénarios de mobilité où les chemins de routage de bout en bout peuvent ne pas être continus, mais ils sont soit statiques (contacts planifiés), soit ils peuvent être découverts avec une probabilité élevée (contacts probabilistes). Cependant, lorsque l' aléatoire dans les contacts des noeuds augmente (contacts opportunistes), la difficulté à découvrir des itinéraires valides vers les destinations du contenu rend plus compliqué de garantir la livraison des données, et encore moins dans des délais précis.

Afin d'augmenter la probabilité de délivrance du contenu dans des réseaux opportunistes, une pratique habituelle dans la littérature DTN est d'utiliser des schémas de routage à copies multiples (multiple copy routing), au lieu de copie unique (par exemple [10], [11]). Dans ce contexte, différents chemins de routage peuvent être créés de manière aléatoire, dans le but de que l'une des copies rencontre la destination. Il est évident que cette approche peut également diminuer le délai de livraison: c'est-à-dire le temps écoulé jusqu'à la délivrance de la première copie de message. Cependant, le suivi des plusieurs chemins et / ou sous-chemins, afin d'assurer la fiabilité, serait plus compliqué et nécessiterait maintenant plus d'informations de contrôle. En outre, si on considère les environnements des ressources limitées (par exemple, les réseaux des capteurs sans fil limitée d'énergie / stockage [12], les réseaux des véhicules avec des durées de contact limitées et / ou limitées par rapport au stockage [13], DTN militaires [14]), la diffusion des plusieurs copies par message dans un réseau DTN peut augmenter la charge globale du trafic dramatiquement, par rapport aux ressources disponibles.

7.2 Motivation

Dans le cadre des réseaux des contacts opportunistes, une pratique courante dans la littérature est de combiner des schémas de routage à copies multiples avec des algorithmes d' ordonnancement et gestion de la congestion (buffer management) local. L'objectif est de décider de façon optimale la quantité des données total et lesquels doivent être répliquées, pendant des occasions des contacts limitées, et comment de traiter les congestions de stockage (par exemple, [15], [16]).

Bien qu'elles ne garantissent pas la fiabilité selon la définition susmentionnée, des telles approches visent à maximiser la performance (par exemple, minimiser le délai de livraison, maximiser le débit de livraison), étant donné un ensemble des contraintes des ressources.

Néanmoins, tels schémas supposent généralement la contention des ressources parmi des sessions des données d' importance égale. Cependant, dans des nombreux scénarios DTN envisagés, les noeuds mobiles devraient lancer plusieurs applications en parallèle. Dans ce contexte, assurant la livraison des données, ou minimisant le délai de la livraison peut être plus important pour une application DTN que pour une autre. Ainsi, plusieurs classes des exigences QoS peuvent être définies et existantes en parallèle. Assurant que les classes individuelles sont satisfaites, lorsque les ressources disponibles le permettent, peut être considérée comme un "lâche" équivalent de la fourniture de fiabilité de QoS dans les réseaux continuellement connectés de bout en bout, où les exigences respectives doivent être capturées toujours.

La spécification de protocole "Bundle" [30] et ses extensions [31] fournissent déjà le cadre pour supporter simultanément différentes classes d'application QoS. Cependant, il n'est pas claire comment de prendre des décisions de prioritisation entre bundles des classes diffèrentes. Un schéma simple, par exemple, pourrait donner une priorité absolue aux applications des classes supérieures. Néanmoins, si la priorisation est basée uniquement sur la classe QoS, les applications appartenant aux classes inférieures seraient "affamés" (c'est-à-dire qu'elles seraient toujours les dernières à être planifiées et les premières à être abandonnées).

Dans ce contexte un nombre de propositions récentes appliquent priorisation entre les classes QoS, essentiellement en distribuant les ressources disponibles (par exemple le nombre de copies par classe, la durée de contact disponible, l'espace tampon disponible par classe) proportionnellement à l'importance de chaque classe QoS [79], [80]. Cependant, cette répartition est basée sur l'application des seuils fixes. Cela soulève un nombre des préoccupations:

- Tout d'abord, il n'est pas clair comment ces seuils pourraient être configurés par rapport à l'environnement d' application.

- Deuxièmement, les seuils fixes ne peuvent pas suivre l'évolution dynamique d'un environnement DTN.

- Enfin, selon la disponibilité des ressources et les paramètres de seuil, le comportement des telles strategies peut être qualitativement différent.

7.2.1 Scènarios des cas d'utilisation

Comme décrit précédemment, la fourniture de QoS efficace pour plusieurs classes des applications lancées simultanément peut être une exigence importante dans des scénarios des cas d'utilisation différents. Il faut cependant souligner que tels cas d' utilisation peuvent se référer aux scénarios complètement déconnectés. En conséquence, les modèles de mobilité peuvent également être très différents. Dans ce contexte, il est important de trouver des solutions fiables pour fournir de QoS, qui peuvent être branchées dans divers cas d'utilisation de DTN et satisfaire les exigences respectives. Dans ce but, nous soulignons ici deux cas d'utilisation distincts qui pourraient bénéficier des telles strategies génériques.

7.2.1.1 Réseaux militaires

Dans le cadre du projet MIDNET (Military Disruption Tolerant Networks) de l'Agence européenne de défense (EDA), nous nous sommes concentrés sur des scénarios militaires. Le but de MIDNET était de proposer des solutions basées sur le paradigme DTN, afin de maintenir les sessions des données actives pendant des déconnexions, tout en assurant une transition transparente vers la connectivité IP, lorsque les conditions du réseau le permettent. Parmi les besoins opérationnels, il est nécessaire de pouvoir prioriser un certain trafic des données par rapport aux autres, sous les contraintes des ressources susmentionnées (c'est-à-dire le stockage, l'énergie, la quantité et la durée des possibilités de communication).

Dans les réseaux militaires, plusieurs classifications pour différents types des services peuvent exister, par rapport à:

- Leur nature (par exemple, communications vocales, diffusion vidéo, messagerie).
- Leur objectif (par exemple, les données des commandes, la connaissance de la situation partagée (shared situation awareness), les rapports de la situation [17]).
- Leur criticité (c'est-à-dire, temps réel, temps non réel mais temps critique, non temps réel-priorité basse, meilleur effort, comme défini dans [18]).

Basé sur des exigences differentes par type de trafic, le classification générique affiché à la figure 7.1 a été fait dans [17] pour représenter les prioritès relatives. Dans ce contexte, la priorité de rétention (retention priority) se réfère à la criticité de maintenir les données respectives stockées aux noeuds DTN (c'est-à-dire, correspondant aux exigences de ratio de délivrance); La priorité de transmission (transmission priority) capture l'importance d'une livraison rapide (c'est-à-dire, correspondant aux exigences de délai de livraison).

Dans ces considérations, multiples classes de priorité de transmission et de rétention ont été définies pour supporter le cadre des applications dans MIDNET [17]. Dans ce contexte, un cadre de priorisation de la qualité de service a été envisagé de fonctionner aux deux niveaux. Le premier niveau se réfère aux strategies statiques qui "filtrent" les messages que devraient en tout cas être priorisés (par exemple messages d'urgence) ou supprimés en premier (par exemple dernières mises à jour d'emplacement remplacent les anciennes mises à jour). Le deuxième niveau se réfère aux strategies dynamiques qui visent à éviter la "famine" des classes de priorité inférieure, tout en respectant les normes QoS des classes de priorité plus élevée, dans les environnements des ressources limitées.

7.2.1.2 Remontée de données FCD dans les réseaux de véhicules hybrides

Dans le cadre d'un contrat de recherche externalisé avec Orange Labs, nous avons étudié l'applicabilité des solutions DTN dans les réseaux des véhicules hybrides et proposé un cadre pour le cas d'utilisation du transfert de FCD (Floating Car Data) via les noeuds d' Infrastructure résidant au bord du réseau (c'est-à-dire les RSUs de communication dédiée à courte portée (Dedicated Short Range Communication (DSRC)), ou les stations de base cellulaires). Les applications FCD se réfèrent généralement à la collection des grandes quantités des données très dynamiques, provenant des vèhicules. Ces données peuvent se référer aux informations de localisation (position, vitesse, direction du mouvement) utiles pour la gestion du trafic [19], ou des données des capteurs, utiles pour des opérations de maintenance et la collection des statistiques

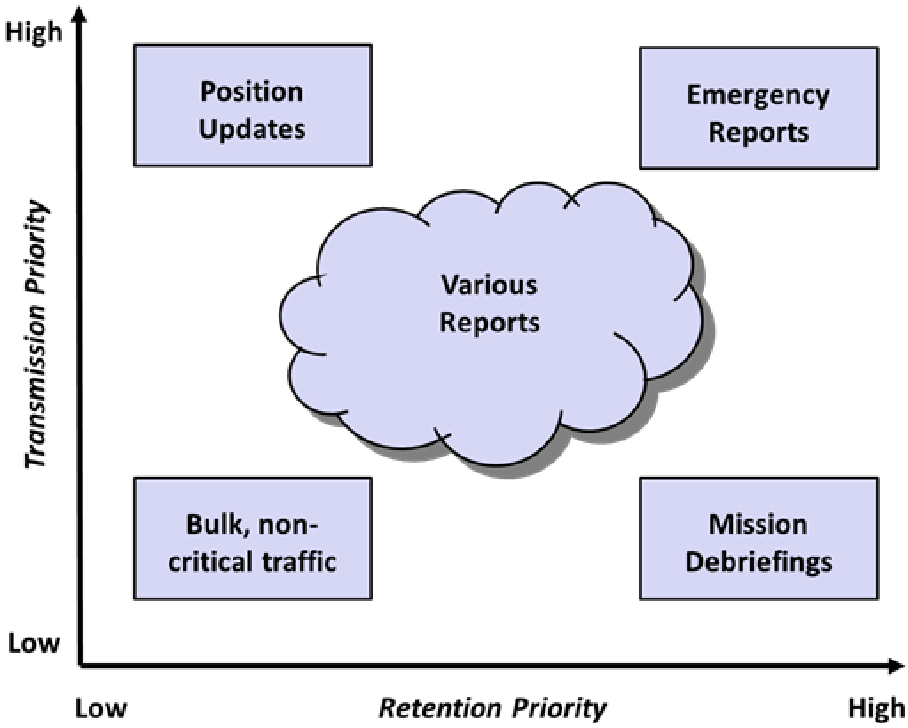

Figure 7.1: Exemples des types de trafic avec différentes priorités de transmission et de rétention

pour les constructeurs automobiles [20]. Donc, les applications FCD peuvent bénéficier du cadre DTN pour survivre pendant des perturbations en raison de n' importe quel des événements suivants:

- Perte temporelle de la connectivité, pendant être à la couverture du même noeud d' Infrastructure.
- Perte de connectivité plus longue en passant de la couverture d'un noeud d' Infrastructure à un autre disponible.
- L'Infrastructure locale est surchargée et ne peut plus accepter de FCD. Dans ce cas, une perturbation "artificielle" est causée.

Compte tenu les lourdes charges du trafic des données, envisagées d' originer des types d'application différents lancés sur les véhicules (par exemple, gestion du trafic, sécurité), les applications FCD devraient produire un type de trafic non critique. Toutefois, si les ressources le permettent, des strategies de priorisation sont nécessaires pour prévenir leur famine, que ce soit en cas de concurrence avec des autres types d'application ou en cas de concurrence interne entre applications FCD différentes.

7.2.2 Les défis de la mobilité dans le monde réel

Dans le contexte du routage à copies multiples et des DTNs opportunistes, ils existent des approches d' ordonnancement et de gestion de stockage qui sont basées sur leurs prédictions par message pour la métrique d'intérêt (par exemple [15], [16]). Donc, une strategie simple pourrait favoriser les messages dont la probabilité de livraison estimée est inférieure à d'autres, par exemple. Cependant, pour faire des prédictions exactes, il est important de capturer les modèles de contact entre les paires des noeuds distincts de manière adéquate. Pour clarifier ça, considèrez que le délai de livraison d'un message peut être exprimé comme le temps nécessaire pour le prochain contact d'un des noeuds portant une copie, avec la destination.

Une approche populaire, dans les protocoles de routage DTN opportunistes, considére que les temps inter-rencontre (inter-meeting times) caractérisant chaque paire des noeuds $< i, j >$ sont indépendants les uns des autres. Dans ce contexte ils peuvent être modélisés par la distribution exponentielle avec un paramètre de fréquence commun $\tilde{\lambda}$, approchant les taux des rencontres individuels $\lambda_{i,j}$ (en supposant des réseaux des contacts homogènes). L'argument de l'exponentialité est soutenu par des études existants, montrant que beaucoup des modèles de mobilité et des traces rèelles correspondent aux modèles des contacts avec des queues approximativement exponentielles [21], [22], [23]. Cependant, il est aussi montré que les distributions des temps inter-rencontre, venant des traces réelles populaires, contiennent des composantes de la loi de puissance. En outre, dans la mobilité réelle, on doit anticiper une grande variance des taux des rencontres entre des paires differentes (réseaux des contacts hétérogènes) [24], [25], [26]. Enfin, certaines paires des noeuds peuvent ne jamais se rencontrer pendant la durée d'une trace de mobilité, conduisant aux graphes des contacts peu connectés. Dans des tels scénarios, le modèle de mobilité exponentielle et homogène susmentionné peut conduire aux estimations inexactes conçernant les temps entre rencontres.

7.2.3 Contributions et Plan de la Thèse

Le but principal de cette thèse est de fournir des garanties de performance QoS pour différentes classes de trafic des données dans le contexte DTN, lorsque les ressources disponibles le permettent. Les exigences de QoS sont associées soit au ratio ou au délai de délivrance. Dans ce sens, comme nous l'avons mentionné précédemment, le cadre de solution que nous appliquons peut être considéré comme un moyen de fournir un "lâche" équivalent de la fourniture de QoS et de la fiabilité dans les réseaux connectés de bout en bout. Nous nous concentrons sur le cas des contacts opportunistes, qui, à l'exclusion des IPNs, caractérisent la majorité des scénarios de mobilité réelle, où la fonctionnalité DTN est utile et, en même temps, les plus stimulants. Dans ce contexte, notre cadre de fourniture de QoS est basé sur des mèthodes distribués d' ordonnancement et gestion de la congestion des tampons. Les schèmes que nous proposons sont "open-loop" (c'est-à-dire ils n' utilisent aucun mécanisme d'accusé de réception des données) et ils utilisent des décisions locales, basées sur des prédictions de livraison pour optimiser la performance globale du réseau. En particulier, nous considérons qu'une allocation optimale des ressources (limitées):

- *S'assurera que les exigences individuelles de qualité de service sont satisfaites lorsque cela est faisable par rapport à la disponibilité des ressources.*
- *Allouera les ressources restantes de façon optimale, afin de maximiser la métrique de performance souhaitée.*

Un aperçu de la thèse est fourni ci-dessous et les contributions réalisées dans chaque chapitre sont résumées.

Chapitre 2 - Etudes Connexes

Dans ce chapitre, nous présentons au debut l'architecture générique qui a été proposée par la communauté de DTNRG [29] pour soutenir les principes de base de DTN. Ensuite, nous nous concentrons sur les aspects de contrôle de la congestion et de la fiabilité dans DTNs, montrant tout d' abord la dépendance entre chaque cadre de solution et l'environnement de mobilité où il est appliqué; plus particulièrement les modèles des contacts des noeuds, comme indiqué précédemment. Nous examinons les systèmes existants dans la littérature, sur la base des plusieurs critères et classéfier les approches que ciblent les réseaux opportunistes (Fig. 7.2). Par ce procedure, nous justifions le cadre général que nous avons utilisé, pour traiter les problèmes de priorisation entre différent classes de qualité de service, pour les scénarios d'intérêt.

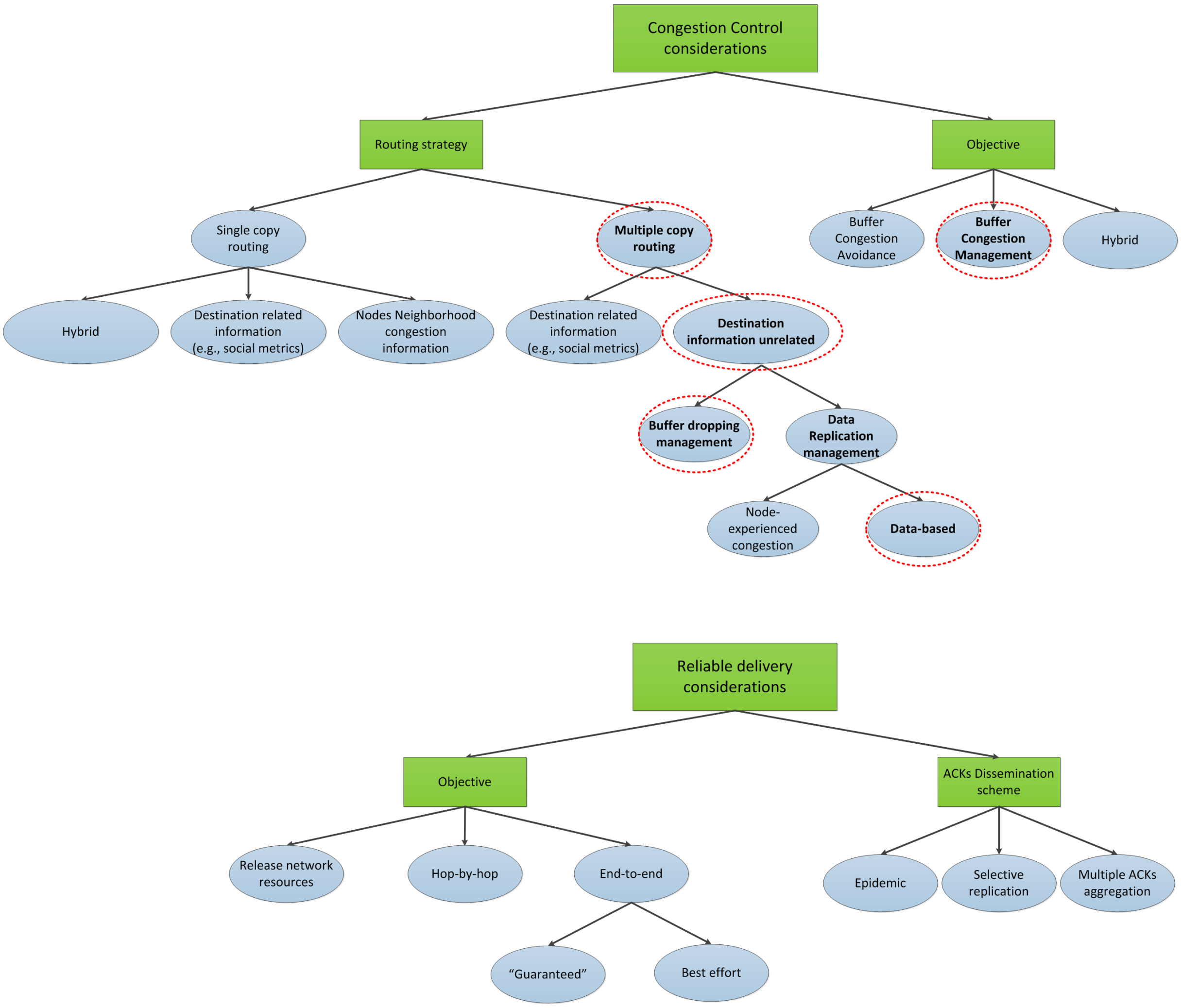

Figure 7.2: Taxonomie des méthodes pour le controle de la congestion et la fiabilité aux DTN, indiquant en rouge nos choix de conception pour la priorisation QoS des données

Le travail relatif à ce chapitre sont présentés dans:

- *P. Matzakos, C. Bonnet, "A taxonomy of congestion control and reliability approaches in opportunistic DTNs", Research Report RR-16-323, Eurecom, September 2016.*

Chapitre 3 - Scénarios des cas d' utilisation

Les scénarios des cas d'utilisation souligné à la section 7.2.1 sont examinés ici en détail. Particulièrement, nous nous concentrons sur leurs contextes d'application distincts et les considérations de mobilité correspondantes, indiquant comment le cadre de priorisation QoS que nous indiquons au chapitre 2 peut être exploité. Dans ce contexte, nous proposons également un ensemble des extensions architecturales DTN, au-dessus des architectures existantes pour chaque cas d'utilisation, afin de soutenir notre cadre dans une perspective de mise en oeuvre pratique.

Plus précisément, pour le cas d'utilisation militaire (MIDNET), la figure 7.3 illustre une architecture basée sur DTN, qu' intégre des modules de routage liés à la provision de QoS. Cette architecture [85] compte sur le modèle proposé par DTNRG [29], combinant les blocs fonctionnels introduits par le protocole Bundle [30] (c'est-à-dire l'agent de protocole bundle) et le protocole de routage PRoPHET [67] (c'est-à-dire, l' agent de protocole de routage et le bloc "Neighbor Discovery"). Bien que nos strategies QoS d' ordonnancement et gestion de la congestion sont indépendants des fonctionnalités de routage du PROPHET, ses blocs de construction et leurs interfaces peuvent intégrer l' intelligence de nos schèmas, en coopération avec quelques mécanismes de soutien (par exemple "Neighbor Discovery").

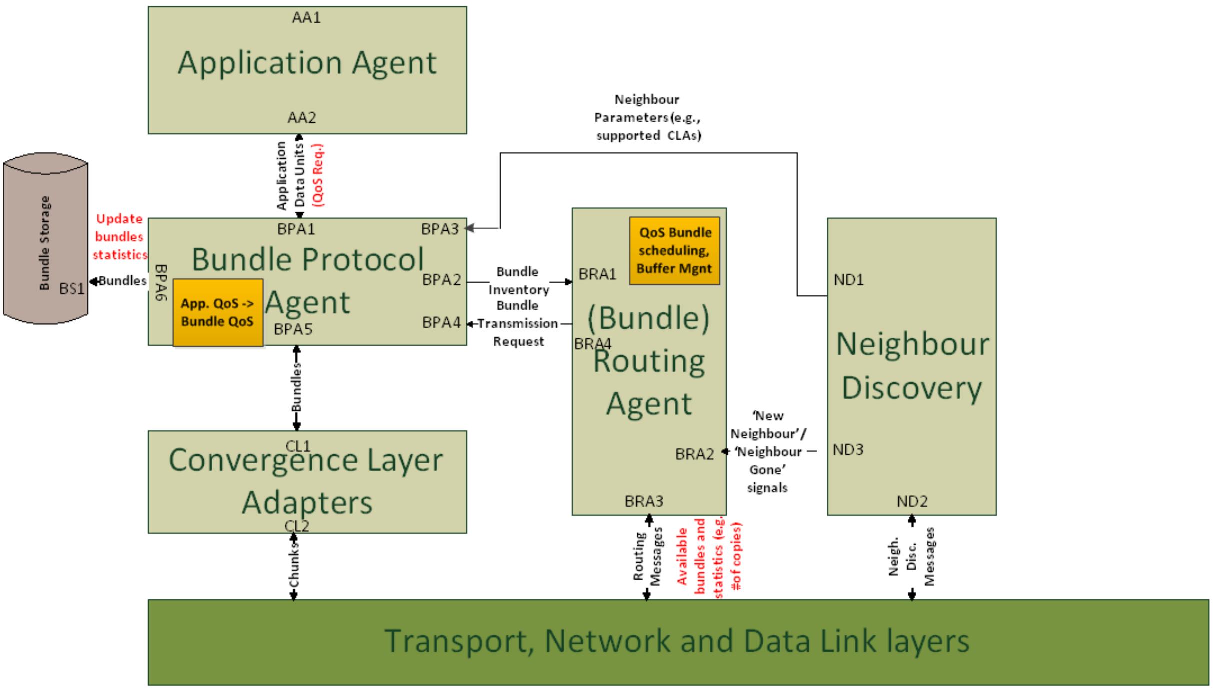

Figure 7.3: Architecture MIDNET QoS

Conçernant le cas d' utilisation véhiculaire, nous suggérons des extensions génériques de l'architecture ETSI pour les systèmes de transport intelligents (ITS) [82], afin de fournir une fonctionnalité hybride. Cette fonctionnalité doit prendre en charge la commutation dynamique entre les modes DTN et non-DTN (délivrance directe), selon les ressources et la disponibilité des opportunités de communication. Sur la base de cette architecture, nous proposons un placement

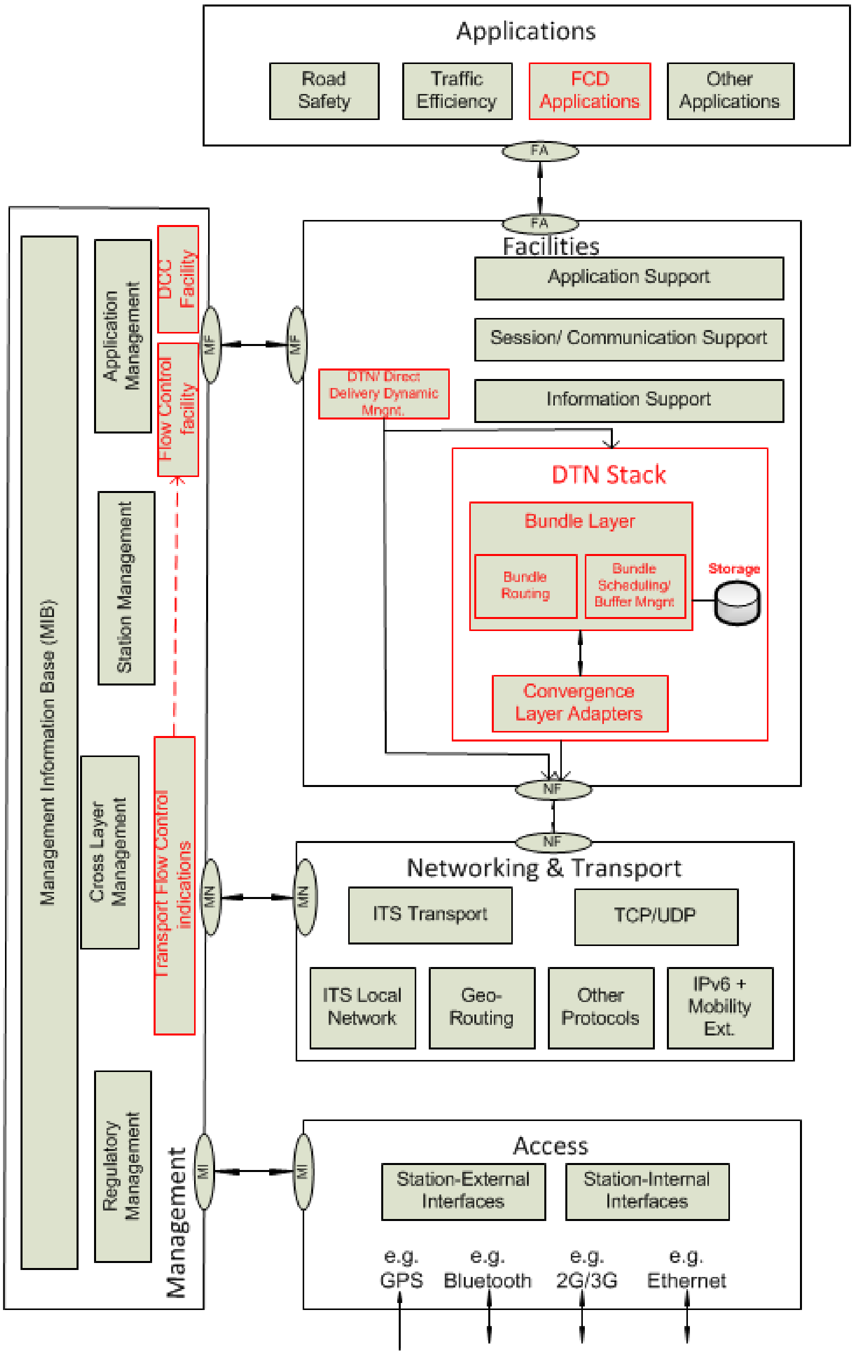

Figure 7.4: Extensions d'architecture ETSI ITS pour supporter la fonctionnalité désirée des DTN et QoS

des blocs DTN appropriés, intégrant nos strategies de priorisation QoS.

Chapitre 4 - Strategies conjointes d' ordonanncement et gestion de la congestion pour des Applications DTN des classes de trafic différentes

Nous formulons d'abord les problèmes de maximisation de la performance du réseau, par rapport à chaque métrique d'intérêt (c'est-à-dire maximiser le débit, ou minimiser le délai moyen de livraison), comme des problèmes d'optimisation sous contraintes convexes. Les contraintes correspondent aux exigences des classes QoS distinctes et aux limitations des ressources. Les contraintes QoS peuvent étre exprimé à la base de chaque message, soit comme une probabilité de délivrance minimum, ou comme un délai de délivrance maximum souhaitée. Une solution centralisée de ce problème serait basée sur des estimations de livraison de chaque message (probabilité ou délai) circulant au réseau, pour effectuer une allocation d'espace tampon parmi les copies des messages non livrés qui est faisable (feasible solution), c'est-à-dire: (i) qu'au moins l'exigence de livraison de chaque message est captée et (ii) optimale, c'est-à-dire qu' il conduit à la maximisation de la performance par rapport à chaque métrique. En particulier, on obtiendrait un vecteur d'allocation optimal n*, dont les entrées correspondraient au nombre des copies pour chaque message individuel.

Toutefois, dans le contexte des DTNs, une telle approche est impossible parce qu'elle a besoin d'une entité centralisée qui connaitrait et contrôlerait l'état de tous les messages existants, instantanément. A cette fin, nous nous concentrons sur une solution distribuée. Les hypothèses fondomentalles qui sont faits et justifiés sont les suivants: (i) le taux moyen des réunions par paire d' un scénario de mobilité spécifique, $\tilde{\lambda}$, est connu, (ii) la connaissance / estimation des paramètres dynamiques de chaque message (par exemple nombre de copies par message) est également disponible. Dans ce contexte, nous derivons des utilités par message optimales, en etendant celles dérivées dans [16] (Eq. 7.1 - 7.2) pour les problèmes sans contraintes de QoS. En particulier, nous ajoutons des fonctions de pénalité appropriés pour considérer la violation des contraintes de QoS (Eq. 7.3 - 7.4) dans les deux problèmes distincts.

- Utilités optimales pour le problème de la **maximisation du débit de livraison** sans des contraintes de QoS [16]:

$$U_i(DR) = \left(1 - \frac{m_i^{(k)}(T_i)}{N-1}\right) \cdot \tilde{\lambda} R_i^{(k)} exp(-\tilde{\lambda} n_i^{(k)}(T_i) R_i^{(k)}), \tag{7.1}$$

- Utilités optimales pour le problème de la **minimisation du délai de livraison** sans contraintes de QoS [16]:

$$U_i(DD) = \left(1 - \frac{m_i^{(k)}(T_i)}{N-1}\right) \frac{1}{n_i^{(k)}(T_i)^2 \tilde{\lambda}}, \tag{7.2}$$

- Utilités optimales pour le problème de la **maximisation du débit de livraison** avec des contraintes de QoS:

$$U_i^{(k)}(DR) = U_i(DR) \cdot \left[1 + \max\{0, c_k(P_{QoS}^{(k)} - P_i^{(k)}(T_i))\}\right] \tag{7.3}$$

- Utilités optimales pour le problème de la **minimisation du délai de livraison** avec des contraintes de QoS:

$$U_i^{(k)}(DD) = U_i(DD) \cdot \left[1 + c_i(T_i^{(k)}) \cdot \max\{0, c_k(E[D_i^{(k)}] - D_{QoS}^{(k)})\}\right]. \tag{7.4}$$

Dans les expressions ci-dessus, $n_i^{(k)}$ représente le numéro des copies du bundle i que devient du QoS class k; $m_i^{(k)}$ représente le numéro des noeuds qui avaient obtenu une copie du bundle i dans quelque moment, indépendamment si ils le disposent encore ou pas; T_i est le temps passé dépuis la création du bundle i; $R_i^{(k)}$ est le restant TTL pour le bundle i; N est le numéro total des noeuds du réseau; $P_{QoS}^{(k)}$ est l' exigance de class k par rapport a la probabilité de délivrance; $D_{QoS}^{(k)}$ est l' exigance de class k par rapport au délai de délivrance;

Conçernant les expressions (7.3) - (7.4) qui considerent des contraintes QoS, les utilités par message que nous avons derivé sont égales aux utilités classiques, $U_i(DR)$ ou $U_i(DD)$ de Eq. (7.1) et (7.2) respéctivement, si le seuil de performance de livraison est estimé d' être capturé. Sinon, ces utilités sont incrémentés d'un terme proportionnel au déficit de performance de délivrance. c_k est une très grande constante qui assure que les utilités des bundles qui ne satisfont pas leur contrainte seront toujours plus élevées que les utilités des bundles qui les satisfont (pour assurer la convergence uniquement aux solutions faisables).

Basé sur le cadre de notre algorithme distribué, on utilise de routage épidémique [10] et, à chaque rencontre entre deux noeuds, un nombre limité des variables du vecteur des copies n peut être affecté (c'est-à-dire seulement ceux-ci correspondant aux messages qui sont dans l'un des deux tampons des noeuds). Néanmoins, la priorisation des messages par rapport à leur classement d'utilité optimal conduit au gain de performance maximum parmi toutes les directions faisables à chaque étape de décision. Donc il corresponde à une implémentation distribuée d'un algorithme gradient ascente (ou descente, selon l'objectif) qui converge finalement vers la solution optimale.

Nous proposons une implémentation spécifique et simple de notre schèmas distribués qui est basé sur le sortage des tampons des noeuds à la manière montré au figure 7.5. Plus précisément:

- Les bundles résidant dans le tampon d'un noeud (file d'attente) peuvent être séparés aux deux groupes dynamiques: Le premier groupe contient tous les bundles dont la probabilité/délai de distribution prévue n'a pas atteint le seuil de QoS souhaité; Le deuxième groupe se compose des bundles qui ont atteint leur seuil de QoS. Les bundles du premier groupe sont toujours priorisés par rapport aux bundles du deuxième groupe.

- Afin de considérer des scénarios où il n'y a pas assez de ressources pour satisfaire toutes les contraintes individuelles de QoS (infeasible domains), nous proposons aussi la définition des sous-groupes distincts dans le premier groupe prioritaire. Chaque sous-groupe est mappé à une seule classe des bundles non satisfaits. C' est l' équivalent de choisir les constantes de pénalité individuelles c_k, dans les expressions (7.3) - (7.4), de manière que $c_1 >> c_2 >> ... >> c_K$ (où 1 corresponde à la classe ayant la priorité nominale la plus élevée et K à la classe avec la priorité la plus basse), afin que les bundles non satisfaits des classes QoS plus élevées seront toujours priorisés par rapport aux bundles respectives des classes QoS plus faibles. L'objectif est d'assurer la satisfaction des classes QoS dans l'ordre de leur importance nominale (c'est-à-dire du plus haut au plus bas).

- Le classement parmi les bundles du même groupe et sous-groupe est basé sur l'utilité classique U_i.

Nous avons validé l'optimalité de notre approche pour les deux problèmes d' optimisation, basé sur des vastes simulations et comparaisons avec d'autres strategies de priorisation, pour des réseaux des contacts homogènes. Dans ce contexte, nous considérons trois classes de priorité:

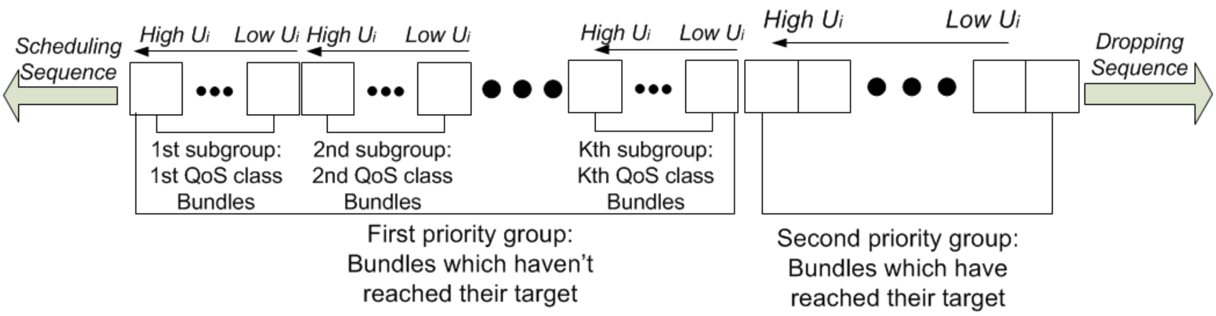

Figure 7.5: Une approche pour l'ordonnancement et la sèquence de blockage des bundles pour les domaines non faisables

"Expedited" (plus élevée), "Normal" et "Bulk" (la plus basse). Les résultats de ratio (BDR) et délai de la livraison (ADD) sont présentés pour diverses valeurs de l'espace tampon total disponible dans le réseau, afin de tester notre strategie, quand nous varions la quantité des congestions de tampon. Les contraintes d'ordonnancement sont appliquées en restreignant le taux moyen des données que peut être échangé par contact au $r_d = 50\%$ du taux moyen non contraint, c'est-à-dire le taux de bundles échangés, s'il n'y a pas de restriction sur le total des données transférées par contact.

Basé sur les figures (7.6) - (7.9), nous soulignons les suivants conçernant la performance de notre strategie:

- Nous vérifions le comportement optimal de notre strategie pour les deux problèmes d' optimisation, par rapport à la disponibilité des ressources. Ainsi, lorsque les ressources ne sont pas suffisantes pour satisfaire toutes les classes QoS (c'est-à-dire le domaine non faisable), les classes QoS sont satisfaites par rapport à la séquence de leur importance. En particulier, nous observons que la classe "Expedited" est la première à satisfaire ses exigences de qualité de service et que, lorsque nous augmentons la disponibilité de l'espace tampon, les autres classes commencent à s'améliorer et finalement elles satisfont leurs contraintes. Pour cette domaine de disponibilité de tampon, la performance de la classe Expedited est stabilisé autour du seuil souhaité. Comme nous augmentons encore les ressources disponibles toutes les classes finissent par avoir la même performance, indiquant la distribution des ressources optimale.

- Conçernant la comparaison de notre schèma avec les autres (c'est-à-dire ORWAR [79] et CoSSD [81] en figures (7.6) et (7.7) respectivement, il est clair que la performance par class de notre strategie est beaucoup mieux alignée avec la performance optimale susmentionné.

- Enfin, il est également clair que notre strategie surpasse les deux autres par rapport à la performance globale dans le domaine faisable des valeurs de tampon (c'est-à-dire de 400 à 800 places).

Le travail de ce chapitre corresponde à la publication suivante:

- *P. Matzakos, T. Spyropoulos and C. Bonnet, "Buffer Management Policies for DTN Applications with Different QoS Requirements," 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015.*

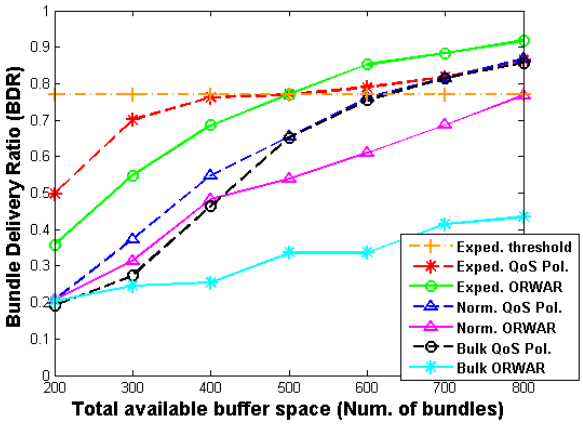

Figure 7.6: QoS Policy vs ORWAR (performance par class)

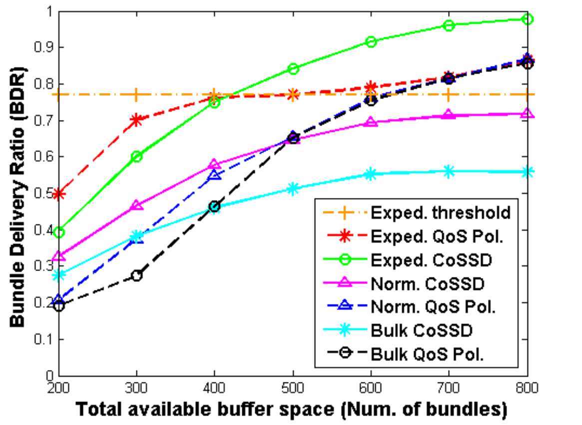

Figure 7.7: QoS Policy vs CoSSD (performance par class)

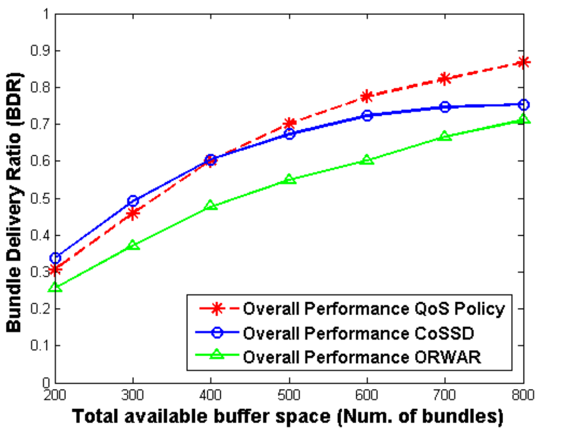

Figure 7.8: QoS Policy vs autres (comparaison globale)

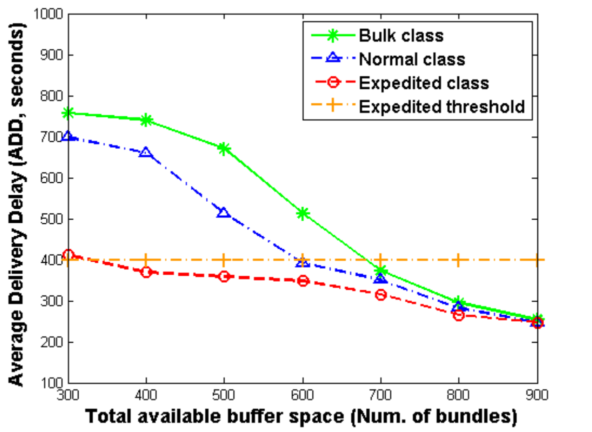

Figure 7.9: Problème d'optimisation ADD, performance par class de la strategie QoS

Chapitre 5 - Amélioration des tèchniques de priorisation QoS pour les réseaux des contacts hétérogènes

En chapitre 4, nous fournissons une solution distribuée et optimale pour le problème de la priorisation QoS dans les DTN aux ressources limitées. Une condition préalable au bon fonctionnement de notre strategie est de faire des prédictions précises conçernant les performances de livraison, par message. En chapitre 4, ces prédictions sont basées sur l'hypothèse que les temps d' interconnexion, entre les paires des noeuds, sont distribué de manière exponentielle et les réseaux des contacts sont homogènes. Dans ce chapitre, nous maintenons l'hypothèse de l'exponentialité, mais nous faisons les extensions nécessaires dans notre cadre de priorisation, afin de considérer des réseaux des contacts hétérogènes et clairsemées, qui correspondent mieux aux conditions de mobilité réelle. Basé sur un cadre spécifique que nous fournissons analytiquement, nous derivons des nouveaux plus précises expressions pour les prédictions de performance (débit et délai de délivrance) et les utilités optimales par message (Eq. 7.5 - 7.10):

- Expression de la probabilité de livraison pour le problème de la **maximisation du débit de livraison aux scénarios de mobilité hétérogènes** (2nd order approximation):

$$\begin{aligned} E_\lambda[P_i^{(k)}(T_i)] = 1 - \Big(1 - \frac{m_i}{N-1}\Big) E[g(\lambda)]^{n_i} = \\ 1 - \Big(1 - \frac{m_i}{N-1}\Big) exp\Big(-\tilde{\lambda} n_i R_i\Big) \Big[1 + \frac{R_i^2 Var(\lambda)}{2}\Big]^{n_i} \end{aligned} \tag{7.5}$$

- Utilités optimales pour le problème de la **maximisation du débit de livraison** sans des contraintes QoS **aux scénarios de mobilité hétérogènes** (2nd order approximation):

$$\begin{aligned} U_i(DR) = \frac{\partial E_\lambda[P_i^{(k)}(T_i)]}{\partial n_i} = \\ \Big(1 - \frac{m_i}{N-1}\Big) \cdot exp\Big(-\tilde{\lambda} n_i R_i\Big)\Big(\tilde{\lambda} R_i - lnA\Big) \cdot A^{n_i} \end{aligned} \tag{7.6}$$

  lorsque $A = 1 + \frac{R_i^2 \cdot Var(\lambda)}{2}$.

- Expression du délai de livraison prévu pour le problème de la **minimisation du délai de livraison aux scénarios de mobilité hétérogènes** (2nd order approximation):

$$E_\lambda\big[D_i^{(k)}(T_i)\big] = \Big(1 - \frac{m_i}{N-1}\Big) \cdot \Big(T_i^{(k)} + \frac{1}{n_i \tilde{\lambda}}\Big[1 + \frac{Var(\lambda)}{\tilde{\lambda}^2}\Big]\Big) \tag{7.7}$$

- Utilités optimales pour le problème de la **minimisation du délai de livraison** sans des contraintes QoS **aux scénarios de mobilité hétérogènes** (2nd order approximation):

$$\begin{aligned} U_i(DD) = -\frac{\partial E_\lambda\Big[D_i^{(k)}(T_i)\Big]}{\partial n_i} = \\ \Big(1 - \frac{m_i}{N-1}\Big) \cdot \frac{1}{n_i^2 \tilde{\lambda}}\Big[1 + \frac{Var(\lambda)}{\tilde{\lambda}^2}\Big] \end{aligned} \tag{7.8}$$

- Limites inférieures et supérieures pour la probabilité et délai de délivrance prevue, respectivement (bound approximations):

$$E_\lambda[P_i^{(k)}(T_i)] \geq 1 - \Big(1 - \frac{m_i}{N-1}\Big) \cdot [EM_2^{DR}(\lambda)]^{n_i} \tag{7.9}$$

$$E_\lambda\Big[D_i^{(k)}(T_i)\Big] \leq \Big(1 - \frac{m_i}{N-1}\Big) \cdot \Big(T_i^{(k)} + \frac{EM_2^{DD}(\lambda)}{n_i}\Big) \tag{7.10}$$

  lorsque $EM_2^{DR}(\lambda)$ et $EM_2^{DD}(\lambda)$ expriment les limites supérieures pour la probabilité de non-délivrance et la délai de délivrance d' une seule copie, respectivement. Ils ont été dérivés en utilisant les bornes supérieures derivés en [92], basés sur l'inégalité d'Edmundson-Madansky pour des fonctions convexes [93].

Si on prend en compte les nouvelles expressions de probabilité et délai de délivrance (Eq. (7.5), (7.7), (7.9) - (7.10)) et les utilités par bundle (Eq. (7.6), (7.8)), nous pouvons les appliquer dans notre algorithme de priorisation QoS, de la même manière que pour le cas des contacts homogènes (Eq. (7.3) - (7.4)).

En outre, nous suggérons une approche alternative basée sur l'utilisation de Spray-and-Wait [11], au lieu de routage épidemique . Il s'agit d'une approche "one-shot" qui, au lieu de surveiller dynamiquement les estimations de performance de livraison par message, sélectionne le nombre maximal des copies par message d'une manière "optimisée" au début de leur durée de vie.

Nous montrons les avantages de performance des notre extensions (approximations "2nd order" et "bound") basé aux simulations avec des traces de mobilité réelles (Cabspotting [94] et Infocom [95]). Au debut, nous comparons la performance de nos extensions avec la mise en oeuvre initiale basé sur le modèle des contacts homogènes:

- **Problème d'optimisation BDR:** Bien qu'il n'y a pas des avantages de performance en ce qui concerne la métrique BDR (Fig. (7.10) - (7.11)), ces avantages sont évidents quand on regarde la métrique ADD de la classe "Expedited" (Fig. (7.13) et (7.15)), quand même au contexte du problème d'optimisation BDR. Ceci est du aux prédictions plus conservatrices des approximations d'ordre supérieur ("2nd order" et "upper bound") qui conduisent à la distribution de plus des copies au début de la durée de vie des bundles (Fig. (7.12) et (7.14)).

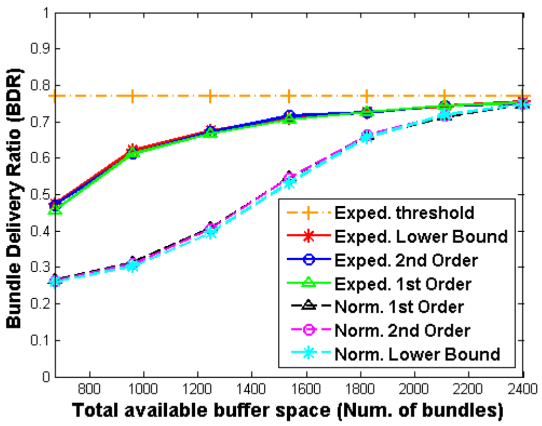

Figure 7.10: Infocom: Optimisation BDR, comparaison approx.

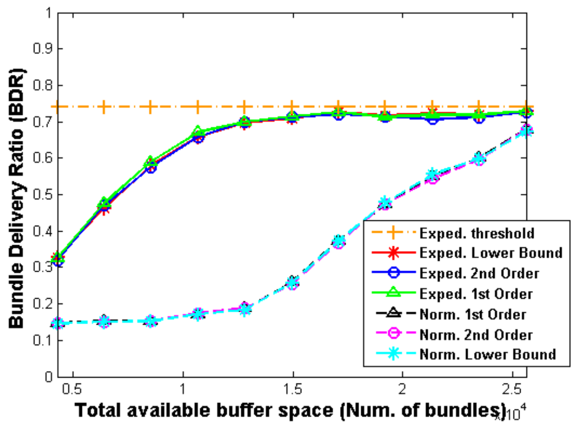

Figure 7.11: Cabspotting: Optimisation BDR, comparaison approx.

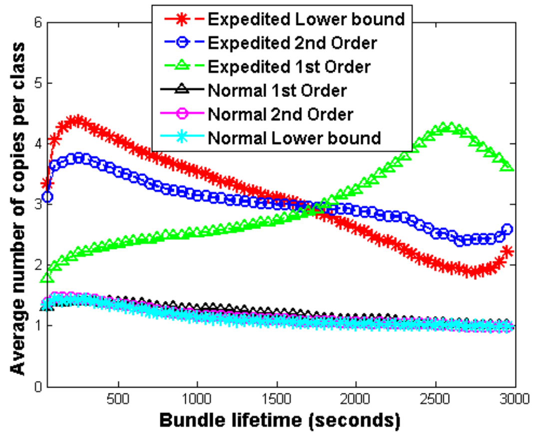

Figure 7.12: Infocom: Copies au fil du temps, comparaison approx. (Buff=10)

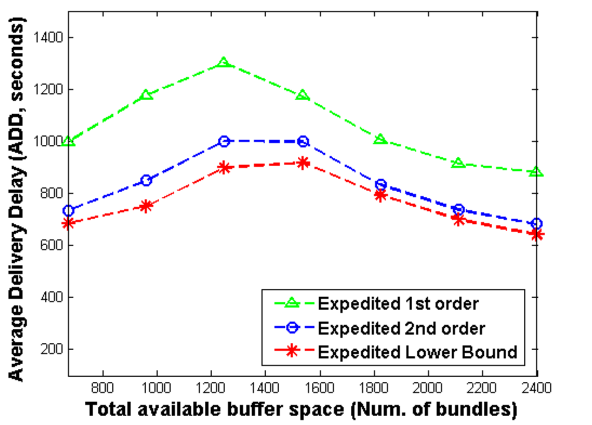

Figure 7.13: Infocom: Optimisation BDR, métrique ADD, comparaison approx.

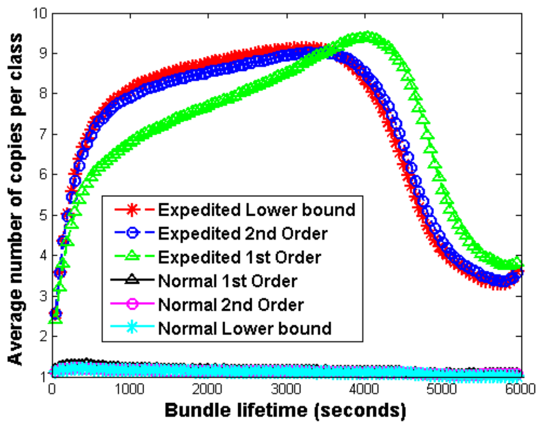

Figure 7.14: Cabspotting: Copies au fil du temps, comparaison approx. (Buff= 20)

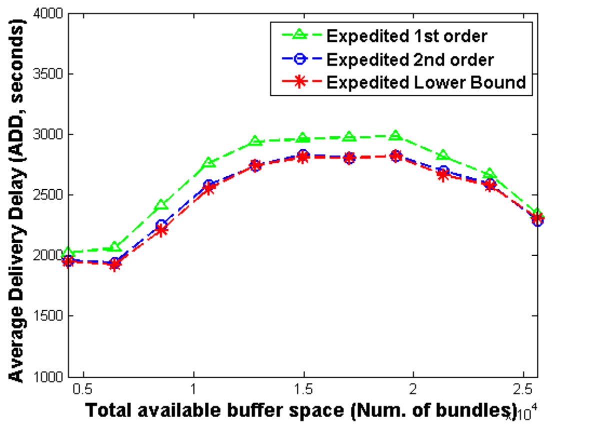

Figure 7.15: Cabspotting: Optimisation BDR, métrique ADD, comparaison approx.

- **Problème d'optimisation ADD:** Les deux approximations plus conservatrices ("2nd order" et " Lower bound") capturent mieux l'exigence QoS de la classe "Expedited", pour toute la gamme des valeurs tampon (et les deux traces), par rapport à l'approximation du premier ordre ("1st order"), dont les mauvaises prédictions ne permettent pas de le faire (Fig. (7.16), (7.18)). Bien sur, la distribution de plus des copies à la classe "Expedited", laisse moins des ressources à la classe "Normal" (Fig. (7.17), (7.19)) et, par conséquent, la performance de ce dernier est mieux avec l'approximation de premier ordre qu'avec les deux autres, pour une gamme des valeurs de tampon. Cependant, nous vérifions le comportement prévu de l'algorithme: quand la disponibilité du tampon augmente, le délai de livraison de la classe "Normal" diminue constamment avec les approximations conservatrices, tandis que la classe "Expedited" reste stable autour de son seuil.

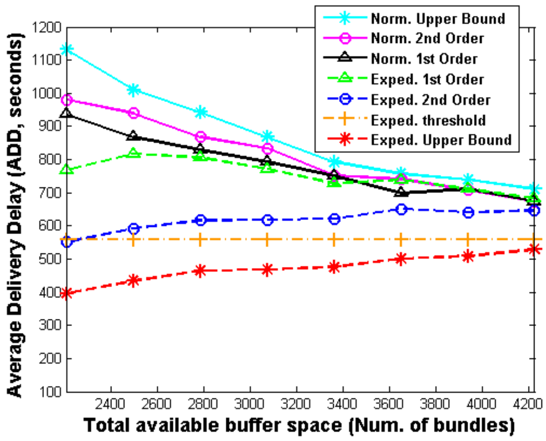

Figure 7.16: Infocom: Optimisation ADD, comparaison approx.

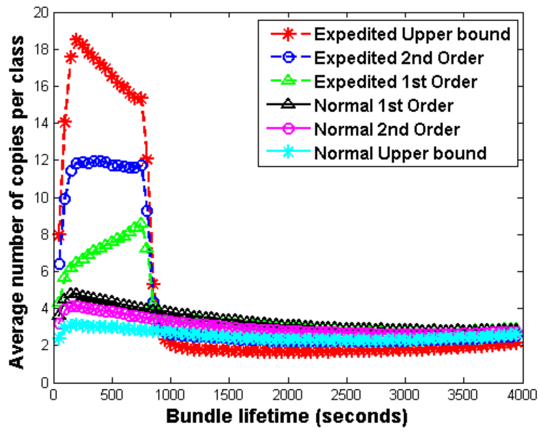

Figure 7.17: Infocom: Copies au fil du temps, comparaison approx. (Buff= 23)

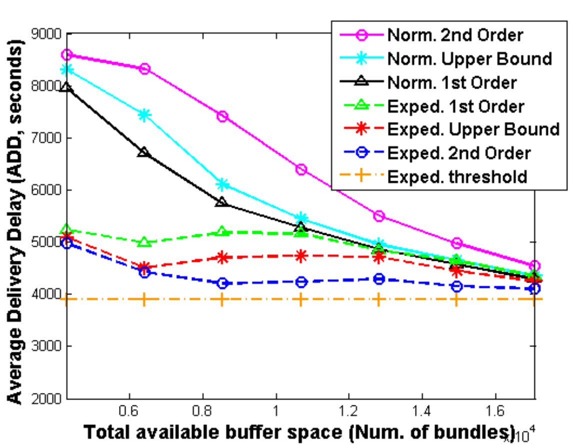

Figure 7.18: Cabspotting: Optimisation ADD, comparaison approx.

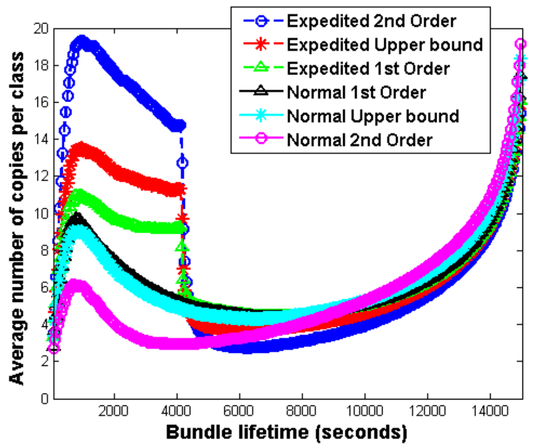

Figure 7.19: Cabspotting: Copies au fil du temps, comparaison approx. (Buff= 20)

- **Comparaison avec ORWAR et CoSSD:** Comme dans le cas des simulations basées sur la mobilité homogène, nous vérifions que notre strategie est beaucoup mieux alignée avec la performance par classe désiré, comparée à ORWAR et CoSSD (Fig 7.20 - 7.21).
- **Comparaison avec la solution alternative basé au Spray-and-Wait:** Bien que la performance de cette strategie est pire que notre strategie de base, comme montré par les résultats de simulation (Fig. 7.22 - 7.23), nous prétendons que cette approche peut être intéressante en raison de sa facilité d'utilisation, puisqu'elle n'exige pas la disponibilité d'informations relatives aux messages en temps réél.

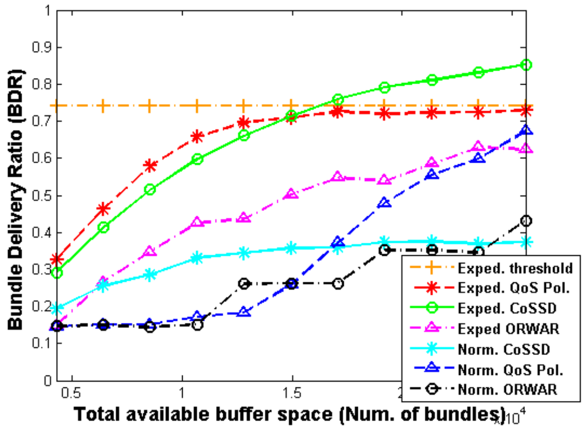

Figure 7.20: Strategie QoS vs ORWAR et CoSSD (Cabspotting)

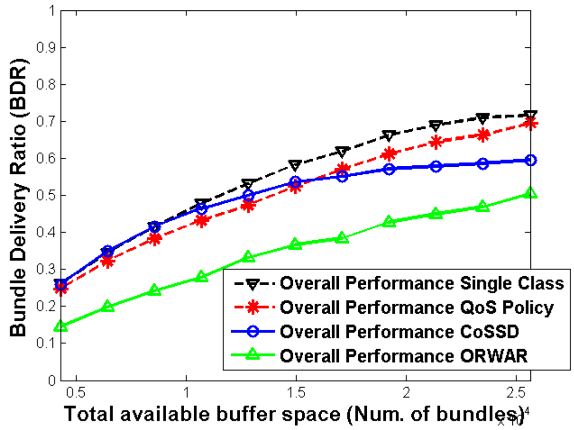

Figure 7.21: Optimisation BDR: Comparaison globale des startegies (Cabspotting)

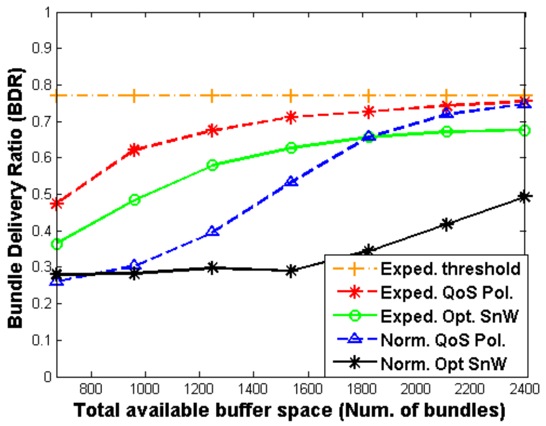

Figure 7.22: QoS Policy vs Opt. SnW (BDR, Infocom)

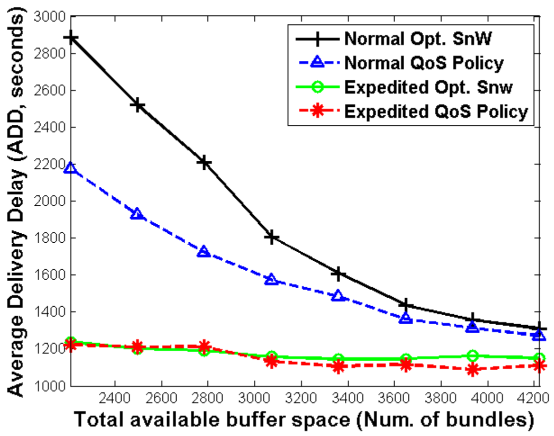

Figure 7.23: QoS Policy vs Opt. SnW (ADD Normalizé, Infocom)

Le travail de ce chapitre corresponde à la soumission suivante:

- *P. Matzakos, T. Spyropoulos, and C. Bonnet, "Joint Scheduling and Buffer Management Policies for DTN Applications of Different Traffic Classes", submitted to IEEE Transactions on Mobile Computing, July 2016.*

Appendix A - Analyse des traces de mobilité réelles basé sur des contacts Pareto et impact sur la performance de priorisation QoS

Dans les chapitres 4 et 5, nous nous sommes basés sur l'hypothèse de l'exponentialité conçernant les distributions des temps inter-rencontre pour les réseaux des contacts homogènes et hétérogènes. Cependant, comme indiqué précédemment, la littérature existante souligne l'existence des fortes composantes du loi de puissance dans ces distributions pour les traces de mobilité réelle. Dans ce contexte, nous supprimons l'hypothèse d'exponentialité et évaluons la

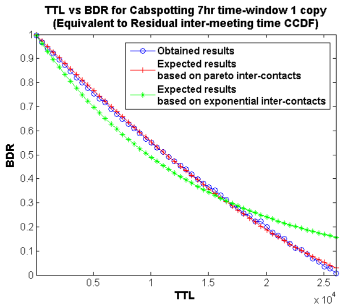

Figure 7.24: Cabspotting: CCDF globale des “residual inter-meeting times”

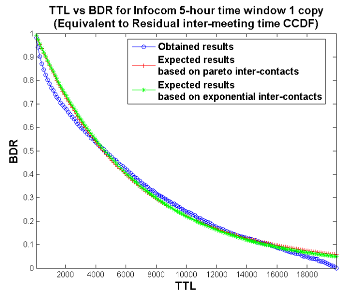

Figure 7.25: Infocom: CCDF globale des “residual inter-meeting times”

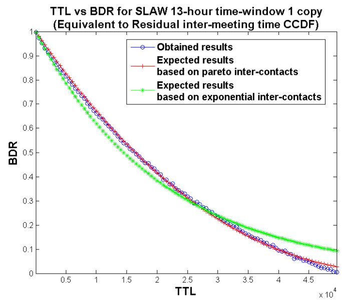

Figure 7.26: SLAW: CCDF globale des “residual inter-meeting times”

performance de notre strategie lorsque le modèle de Pareto généralisé (appartenant à la famille des distributions de la loi de puissance) est utilisé à sa place.

A cette fin, au debut nous décrivons le cadre utilisé pour extraire le type de distribution approprié pour notre but par des traces de mobilité. Nous comparons ensuite les résultats de prédiction obtenus quand on modélise cette distribution par les modèles exponentiel et Pareto generalisé, dans des scénarios sans contraintes des ressources et pour différentes traces de mobilité réelle (Cabspotting[94], Infocom[95] et SLAW[96]). Pour la majorité des traces, il semble que la modélisation basée sur Pareto surpasse celle d' exponentielle homogène (c'est-à-dire, sur la base des formulations du chapitre 4). En effet, aux figures 7.24 et 7.26 on peut observer que, pour les traces Cabspotting et SLAW, la distribution de Pareto généralisée peut décrire les distributions obtenues avec une précision plus élevée que l'exponentielle.

Ensuite, nous nous sommes concentrés à nouveau sur notre problème de priorisation des ressources limitées. Nous avons comparé l'implémentation de notre algorithme basée sur Pareto avec l'exponentielle du 1er ordre (c'est-à-dire sur la base des formulations du chapitre 4) et les extensions exponentielles pour les contacts hétérogènes (c'est-à-dire sur la base des formulations du chapitre 5). Dans le contexte de notre implémentation Pareto, nous avons derivé les expressions correspondant pour les prédictions de performance (débit et délai de délivrance) et les utilités optimales par message (Eq. 7.11 - 7.14):

- Expression de la probabilité de livraison pour le problème de la **maximisation du débit de livraison aux scénarios de mobilité hétérogènes** (Pareto):

$$P_i(T_i) = \left(1 - \frac{m_i(T_i)}{N-1}\right) \cdot \left(1 - \left(1 + \frac{kR_i}{\sigma}\right)^{-\frac{p_s n_i(T_i)}{k}}\right) + \frac{m_i(T_i)}{N-1} \tag{7.11}$$

- Utilités optimales pour le problème de la **maximisation du débit de livraison** sans des contraintes QoS **aux scénarios de mobilité hétérogènes** (Pareto):

$$\begin{aligned} U_i(DR) = \frac{\partial P_i(T_i)}{\partial n_i} = \\ \left(\frac{p_s \cdot (N - m_i(T_i) - 1)}{k(N-1)}\right) \cdot \left(1 + \frac{kR_i}{\sigma}\right)^{-\frac{n_i(T_i)}{k}} \cdot ln\left(1 + \frac{kR_i}{\sigma}\right) \end{aligned} \tag{7.12}$$

- Expression du délai de livraison prévu pour le problème de la **minimisation du délai de livraison aux scénarios de mobilité hétérogènes** (Pareto):

$$E[D_i(T_i)] = \left(1 - \frac{m_i(T_i)}{N-1}\right) \cdot \left(T_i + \frac{\sigma}{p_s n_i(T_i)(1-k)}\right) \tag{7.13}$$

- Utilités optimales pour le problème de la **minimisation du délai de livraison** sans des contraintes QoS **aux scénarios de mobilité hétérogènes** (Pareto):

$$\begin{gathered} U_i(DD) = -\frac{\partial E\left[D_i(T_i)\right]}{\partial n_i} = \\ \left(1 - \frac{m_i(T_i)}{N-1}\right) \cdot \left(\frac{\sigma}{p_s(1-k)n_i(T_i)^2}\right) \end{gathered} \tag{7.14}$$

lorsque les paramètres “shape” k et “scale” σ decrive la distribution Pareto pour chaque trace specifique; p_s est le coefficient de densité de chaque trace (c'est-à-dire le nombre des paires qui se rencontrent sur le nombre des paires de noeuds total qui existent dans le réseau).

Dans ce contexte, il etait intéressant de constater que les résultats comparatifs des différentes méthodes d'approximation ont eu quelques différences qualitatives en ce qui concerne le problème d'optimisation et la trace de mobilité. Particulièrement:

- Pour **le cas d'optimisation BDR**, l'implémentation de Pareto a surpassé l'exponentielle du premier ordre pour tous les scénarios de mobilité évalués (Fig. 7.27 - 7.32). Cependant, en ce qui concerne la comparaison avec les exponentielles d'ordre supérieur, la performance par classe de la mise en oeuvre avec Pareto etait similaire avec celles pour la trace Cabspotting, mais légèrement plus proche de la performance prévue pour le cas de la trace SLAW (Fig. 7.28). Ceci nous a amèné à conclure que la distribution généralisée de Pareto était la méthode d'approximation la plus appropriée pour la trace spécifique, en raison de sa courte variance des taux inter-rencontre entre les paires des noeuds différentes et sa forte composante du loi de puissance.

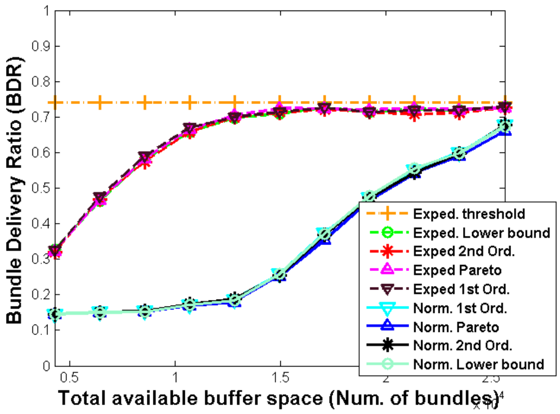

Figure 7.27: Cabsp.: Opt. BDR, comparaison entre approx. Pareto et exponentielles

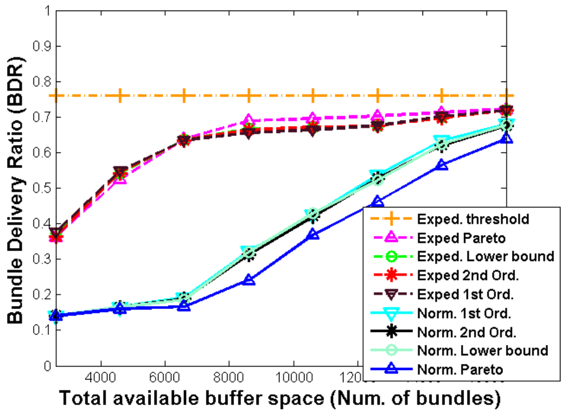

Figure 7.28: SLAW: Opt. BDR, comparaison entre approx. Pareto et exponentielles

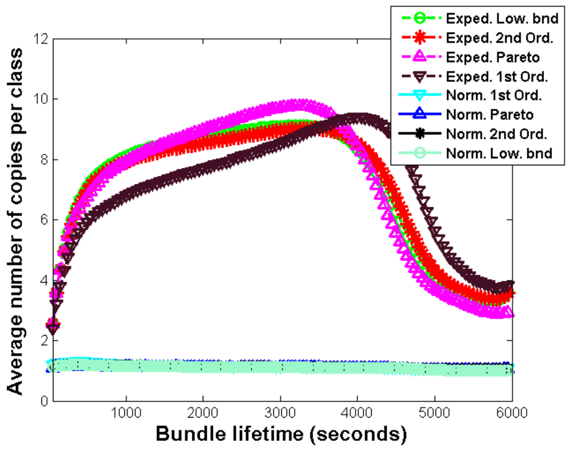

Figure 7.29: Cabspotting: Opt. BDR Copies au fil du temps, comparaison entre approx. Pareto et exponentielles

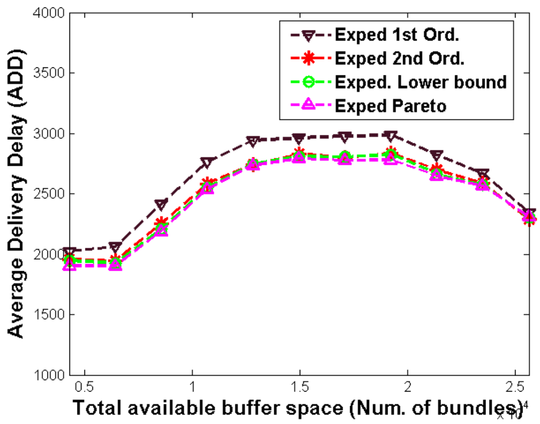

Figure 7.30: Cabspotting: Opt. BDR, ADD métrique, comparaison entre approx. Pareto et exponentielles

- Pour **le cas d'optimisation ADD**, nous avons toutefois observé que le modèle de Pareto de premier ordre n'apportait aucun avantage de performance pour les scénarios évalués, comparativement aux autres méthodes d'approximation (Fig. 7.33 - 7.36).

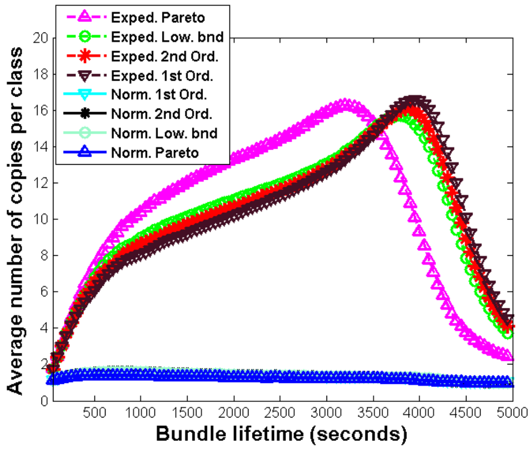

Figure 7.31: SLAW: Opt. BDR, Copies au fil du temps, comparaison entre approx. Pareto et exponentielles

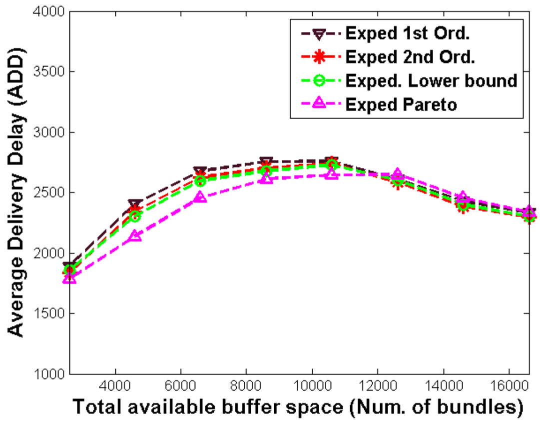

Figure 7.32: SLAW: Opt. BDR, ADD métrique, comparaison entre approx. Pareto et exponentielles

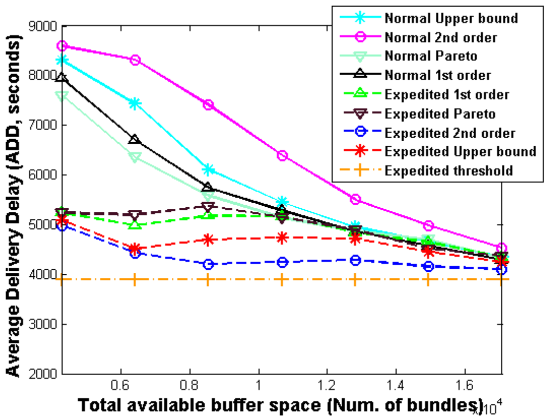

Figure 7.33: Cabspotting: Opt. ADD, ADD métrique: comparaison entre approx. Pareto et exponentielles

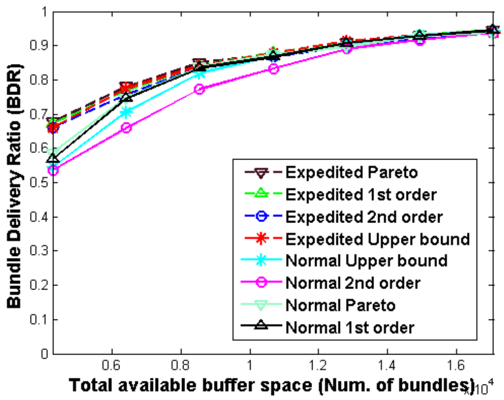

Figure 7.34: Cabspotting: Opt. ADD, BDR métrique: comparaison entre approx. Pareto et exponentielles

Appendix B - Une architecture IPv6 pour les services de mobilité intelligente Cloud-to-Vehicle sur les réseaux des véhicules hétérogènes

Le travail décrit dans ce chapitre est indépendant du reste des contributions de cette thèse. En particulier, nous fournissons la spécification d'une application Point-of-Interest (PoI) initiée par le cloud et nous illustrons ses exigences pour une convergence entre les services géographiques de gestion de la mobilité IPv6 et les services géographiques DSRC (Dedicated Short Range Communications). Nous proposons d' étendre une architecture de gestion de mobilité IPv6 plate avec un nouveau bloc fonctionnel: LIMME (Location and Infrastructure Mobility Management Entity), composé de trois fonctions importantes: un Location Manager (LM) servant comme un point d' ancrage pour les services basés sur le cloud; une fonction pour la gestion de la mobilité géographique (GMM) que serve comme un proxy de localisation pour le LM et traite la mobilité IPv6; un sélecteur de noeud d'infrastructure qui sélectionne un itinéraire des données basé sur

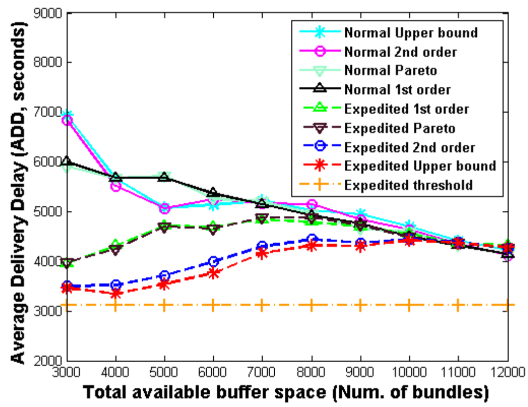

Figure 7.35: SLAW: Opt. ADD, ADD métrique: comparaison entre approx. Pareto et exponentielles

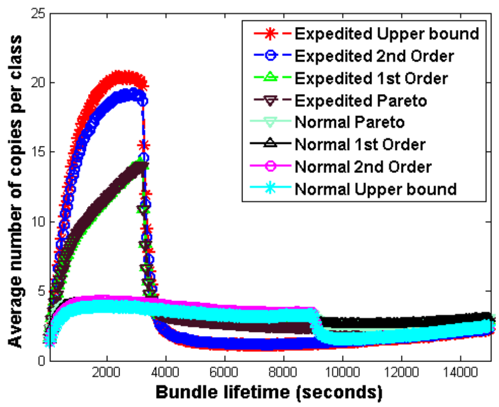

Figure 7.36: SLAW: Opt. ADD, Copies au fil du temps, comparaison entre approx. Pareto et exponentielles

des données géographiques et des conditions des noeuds d'infrastructure locales. Comme preuve de concept, nous avons implémenté ces extensions sur la plate-forme de simulation ITS iTETRIS [28] et illustré leurs avantages dans la gestion de la mobilité IPv6 et le déchargement du trafic.

Le travail de ce chapitre corresponde à la publication suivante:

- *P. Matzakos, J. Härri, B. Villeforceix and C. Bonnet, "An IPv6 architecture for cloud-to-vehicle smart mobility services over heterogeneous vehicular networks," 2014 International Conference on Connected Vehicles and Expo (ICCVE), Vienna, 2014.*

Appendices

Appendix A

Analysis of real mobility traces based on Pareto contacts and impact on QoS prioritization performance

A.1 Introduction

In chapters 4 and 5, we have been based on the assumption of exponentially distributed inter-meeting times, throughout our analysis, to support our QoS prioritization framework for homogeneous and heterogeneous contact networks, respectively. However, existing analysis in the literature, based on real mobility traces, highlights that inter-meeting time distributions often exhibit more of a power-law behavior with (potentially) exponential tails [22], [26]. Among the arguments, it is supported that power-law distributions can be better to describe the heterogeneity encountered in real mobility [26]. To this end, in the current chapter we examine the performance of delivery predictions when, instead of exponential, the inter-meeting times are modeled through the generalized Pareto distribution.

In the context of the algorithm proposed in chapter 4, our aim was to first extract an appropriate type of distribution which considers all the individual pairwise inter-meeting times as equally important. This is straightforward when modeling the inter-meeting patterns through the exponential distribution, thanks to its dependency from a single mean rate parameter $\tilde{\lambda}$, which can be extracted directly from a trace. This is not the case for modeling based on power-law functions though. We first highlight the difference of the desired distribution from the *aggregate inter-meeting times distribution* of a mobility trace and then describe the framework we used to extract it (section A.2).

Then, we compare the predictions performance of the first order exponential and generalized Pareto models, based on a resource and QoS unconstrained evaluation scenario (section A.3). It is shown that, in this simple scenario, the Pareto based predictions are more accurate than the exponential ones. Finally, we go back to our initial QoS prioritization problem and compare the performance of our scheme when its implementation is based either on the generalized Pareto model, or the exponential models (first, second order and bound approximations), in the context of the BDR and ADD optimization problems (section A.4), formulated in chapter 4.

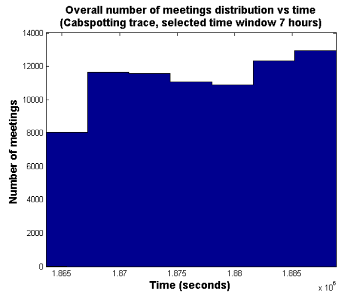

Figure A.1: Cabspotting: overall meetings histogram

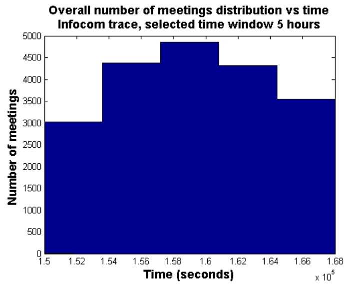

Figure A.2: Infocom: overall meetings histogram

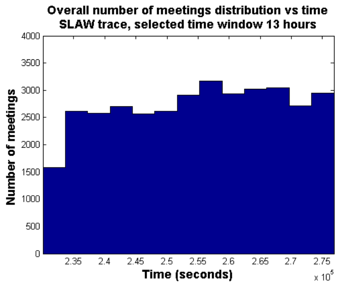

Figure A.3: SLAW: overall meetings histogram

A.2 Methodology

A.2.1 Definitions and context

The *pairwise inter-meeting time distribution* between two nodes refers to the distribution obtained for the inter-meeting time, sampled over each meeting of these two nodes. The *aggregated inter-meeting time distribution*, is the distribution of the inter-meeting time samples, over all distinct pairs of nodes. Finally, the *residual inter-meeting time* refers to the time until the next meeting of a node pair from a given observation time.

In previous chapters we have highlighted the importance of describing accurately appropriate inter-meeting patterns in order to derive delivery predictions for the metric of interest (i.e., delivery ratio, or delay). Based on the system model described in chapter 4 (section 4.2.1), any node participating in a mobility trace has equal probability of being a source or destination of a bundle. As a result, the inter-meeting times statistics of each pair of nodes are equally important in extracting the overall inter-meeting time distribution that we need. In this context, we claim that the aggregated inter-meeting times distribution, although giving a good intuition for the form of the required distribution, is not appropriate as an indicator of a bundle's delivery probability or delay. The reason is that, in the aggregated distribution, the statistics of node pairs which meet more frequently are dominant over the statistics of pairs which meet less frequently.

Another important remark is related to the packet generation context. Based on our framework, a packet generation referring to a particular source-destination pair can take place at any point in time between two successive meetings of the couple. This is compliant with the definition of the residual inter-meeting time that we highlighted before. Thus, if we consider that TTL can be infinitely large, the packet generations can be seen as observation times and the duration until they get delivered to their destination, as residual inter-meeting times.

A.2.2 Extracting the appropriate distribution

Based on the previous discussion, we provide here the methodology to extract an equivalent of the residual inter-meeting times distribution which is appropriate for estimating the metric of interest. Our analysis is based on three different mobility traces, namely: the Cabspotting and Infocom traces, which were also used for the evaluation of our policy in chapter 5, and a trace originating from the SLAW mobility model. SLAW [96] is a state-of-the art mobility model

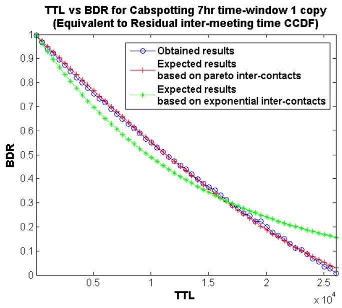

Figure A.4: Cabspotting: Overall residual inter-meeting times CCDF

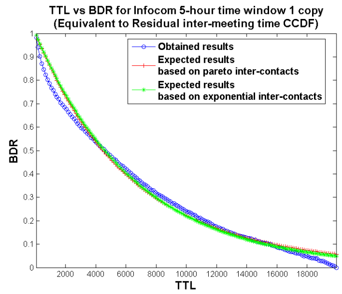

Figure A.5: Infocom: Overall residual inter-meeting times CCDF

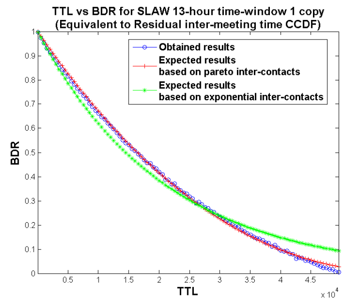

Figure A.6: SLAW: Overall residual inter-meeting times CCDF

which can produce realistic synthetic traces of the human walk. For all the traces, we focus on time windows where we have generally large amount of meetings (daytime) which are relatively uniformly distributed over time. In Fig. A.1 -A.3 the overall meetings histograms per trace are depicted, where each bar corresponds to a one-hour interval.

For each trace, we extract an approximation of the overall residual inter-meeting times cumulative distribution function (CDF) we are looking for, based on the following simulation framework. Each node participating in the trace is accounted as a destination for which packets are generated periodically. The source node of the packets is selected randomly out of the rest of the nodes in the trace and only direct delivery is allowed (i.e., no packet replication) [1]. For each destination node, this simulation process is repeated for different source-destination pairs and for packet TTL values ranging from 50 seconds to the duration of the time window (i.e. 7 hours for Cabspotting, 5 hours for Infocom and 13 hours for SLAW). This is done, in order to capture all the range of possible residual inter meeting time values. At the end, BDR averages are extracted for each value of TTL, out of all source-destination pairs. Obviously, the larger the TTL value the larger the corresponding average BDR. Thus, we end up having a distribution of BDR values corresponding to different TTL values, which approximates the required CDF, as all the source-destination pairs are equally considered in its extraction.

In figures A.4 - A.6 the acquired approximations for the CCDFs (Complementary CDF) of interest for the three different traces are depicted. Based on these results, a curve fitting procedure was performed, following a least squares algorithm for the two probability distributions we examine: i.e., generalized Pareto (Eq. A.1) and the well known exponential (Eq. A.2).

$$\bar{F}_{T_{Par}}(t) = \left(1 + k \cdot \frac{t-\mu}{\sigma}\right)^{-\frac{1}{k}} \tag{A.1}$$

$$\bar{F}_{T_{Exp}}(t) = exp(-\tilde{\lambda} \cdot t) \tag{A.2}$$

Based on this procedure, we extracted the fitting distributions parameters that describe it: i.e. shape k and scale σ for generalized Pareto (considering threshold $\mu = 0$) and rate parameter

[1] A prerequisite for a node to be selected as a source for a packet with a given destination is to meet this destination at least once during the trace's duration.

$\tilde{\lambda}$ for the exponential [2]. Thus, it can be observed that, for the Cabspotting and SLAW traces, the generalized Pareto distribution can describe the obtained distributions with higher accuracy than the exponential. For the case of the Infocom trace, though, the curves corresponding to the two distributions practically coincide. Thus, we will focus our further analysis on the other two traces.

A.3 Performance Evaluation for QoS and resource unconstrained scenarios

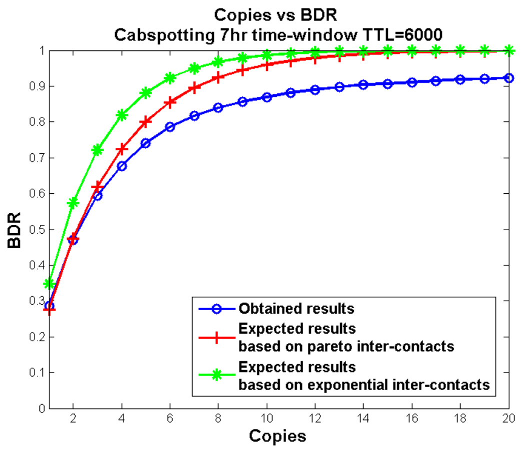

Figure A.7: Cabspotting: Number of copies vs BDR (TTL=6000)

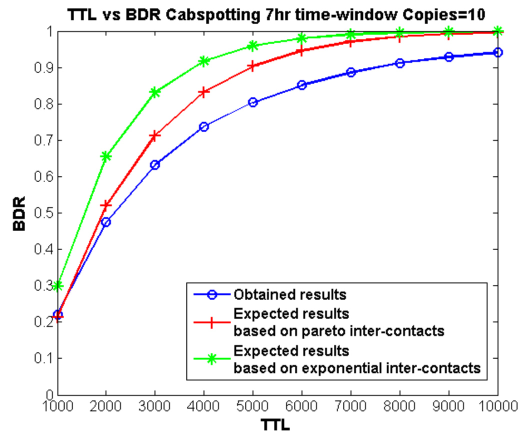

Figure A.8: Cabspotting: TTL vs BDR (10 Copies per bundle)

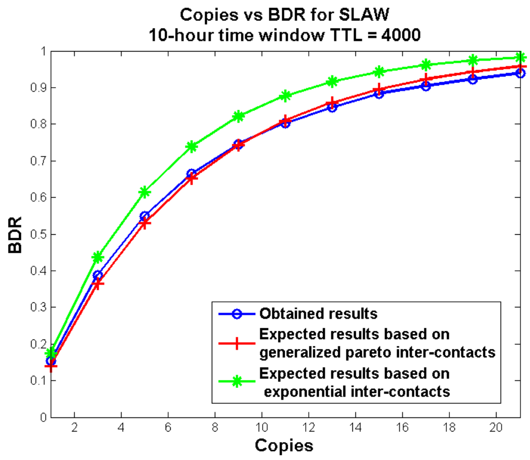

Figure A.9: SLAW: Number of copies vs BDR (TTL=4000)

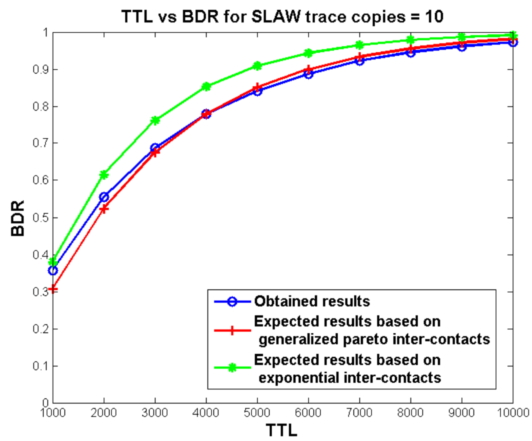

Figure A.10: SLAW: TTL vs BDR (10 Copies per bundle)

In this section, the aim is to evaluate and compare the impact of the exponential and generalized Pareto distributions on the BDR predictions accuracy, as we vary the degree of replication

[2] We note that the extracted rate parameter of the exponential distribution corresponds to the mean pairwise meeting rate.

per bundle but still without applying any type of constraints in the simulation scenarios. Particularly, still only direct delivery transmissions are allowed but now each generated bundle has multiple replicas in the network from the beginning of its lifetime. Thus, the Pareto and exponential based predictions can be expressed as a function of the assigned number of copies n and TTL per bundle as:

$$F_{T_{Par}}(t) = 1 - \left(1 + k \cdot \frac{TTL}{\sigma}\right)^{-\frac{n}{k}} \tag{A.3}$$

$$F_{T_{Exp}}(t) = 1 - exp(-\tilde{\lambda} \cdot n \cdot TTL) \tag{A.4}$$

Once again, the source-destination pairs are selected randomly with the only restriction being that they encounter each other at least once during the duration of the trace. Thus, it can be verified that the better description of the residual inter-meeting times distribution by generalized Pareto (Fig. A.4 and A.6) leads to more accurate predictions of the BDR performance, for varying number of copies per bundle (Fig. A.7 and A.9) and varying TTL (Fig. A.8 and A.10).

A.4 Revisiting the QoS prioritization problem

In the following, we revisit the problem of QoS prioritization. The aim is to compare the performance of the exponential based implementations of our policy, derived in chapters 4 and 5 (i.e., first order, second order and bound approximations), with the implementation based on the generalized Pareto distribution.

We focus on both optimization problems described in chapter 4. For the BDR optimization case, based on the formulation of the problem in section 4.2.2 and the generalized Pareto model for the residual inter-meeting times, the per bundle delivery probability expression is the following (using the same notation with chapters 4 and 5):

$$P_i(T_i) = \left(1 - \frac{m_i(T_i)}{N-1}\right) \cdot \left(1 - \left(1 + \frac{kR_i}{\sigma}\right)^{-\frac{p_s n_i(T_i)}{k}}\right) + \frac{m_i(T_i)}{N-1} \tag{A.5}$$

The objective and constraint functions (Eq. 4.1 and 4.2) can be expressed based on Eq. A.5. Accordingly, the optimal QoS unconstrained utilities can be derived by differentiating Eq. A.5 with respect to the number of copies per bundle $n_i(T_i)$, as:

$$\begin{aligned} U_i(DR) = \frac{\partial P_i(T_i)}{\partial n_i} = \\ \left(\frac{p_s \cdot (N - m_i(T_i) - 1)}{k(N-1)}\right) \cdot \left(1 + \frac{kR_i}{\sigma}\right)^{-\frac{n_i(T_i)}{k}} \cdot ln\left(1 + \frac{kR_i}{\sigma}\right) \end{aligned} \tag{A.6}$$

For the ADD optimization case, following the formulation of section 4.2.2, an estimation on the expected delivery delay, $E[D_i(T_i)]$, is needed. Equivalently to the first order exponential approximation, the mean delivery delay for a single copy can be expressed through the mean value of the generalized Pareto distribution as $\frac{\sigma}{1-k}$. Accordingly, $E[D_i(T_i)]$ can be written as a function of the number of copies per bundle $n_i(T_i)$, the elapsed time since its creation T_i and the number of seen nodes $m_i(T_i)$, as:

$$E[D_i(T_i)] = \left(1 - \frac{m_i(T_i)}{N-1}\right) \cdot \left(T_i + \frac{\sigma}{p_s n_i(T_i)(1-k)}\right) \tag{A.7}$$

The optimal unconstrained utilities are then derived as:

$$U_i(DD) = -\frac{\partial E\left[D_i(T_i)\right]}{\partial n_i} = \left(1 - \frac{m_i(T_i)}{N-1}\right) \cdot \left(\frac{\sigma}{p_s(1-k)n_i(T_i)^2}\right) \tag{A.8}$$

Thus, applying Eq. A.5 - A.8 in the distributed framework described in section 4.2 constitutes the Pareto based implementations of the QoS prioritization algorithm, for the respective optimization problems.

Note that, in Eq. A.5 - A.8, we have also considered their dependency from the network density coefficient, p_s. Given the random selection of bundle source-destination pairs, we claim that the "valuable" number of copies per bundle: i.e., copies that have some probability of encountering the destination before expiry, for some $0 < p_s \leq 1$, can be approximated as: $p_s \cdot n_i(T_i)$. This is the equivalent of the way the mean pairwise contact rate was defined for sparse topologies in section 5.2.4.

A.4.1 Performance comparisons

The evaluation and comparison of the Pareto implementation with the exponential ones are performed based on the selected traffic windows of the Cabspotting and SLAW mobility traces (Fig. A.1 and A.3). The configuration parameters for the two mobility scenarios are summarized in table A.1. As done in chapter 5 (section 5.3.2), we examine both BDR and ADD metrics, in the context of both optimization problems.

	Cabspotting	*SLAW*
Number of Nodes (N)	536	200
Total simulation time (min.)	440	822
Mean pairwise meeting rate ($\tilde{\lambda}$, min^{-1})	$4.1 \cdot 10^{-3}$	$3.1 \cdot 10^{-3}$
Pairwise meeting rates variance (σ^2_{λ}, min^{-2})	$8.9 \cdot 10^{-6}$	$4.33 \cdot 10^{-6}$
Pareto shape parameter k	-0.69	-0.44
Pareto scale parameter σ	$1.94 \cdot 10^4$	$2.74 \cdot 10^4$
Density coefficient (p_s)	47%	76%
Bundle TTL (BDR Optimization problem, sec.)	6000	5000
Bundle TTL (ADD Optimization problem, sec.)	15000	15000
Mean rate of contact window r_d (% of unconstrained rate)	0.5	0.5
Expedited desired QoS (BDR opt.)	0.74	0.76
Expedited desired QoS (ADD opt, sec.)	3900	3120

Table A.1: Simulation Parameters for the comparison with the Pareto based implementation of the QoS policy

A.4.1.1 BDR Optimization

In terms of the BDR metric, no difference is observed in the per class performance between the Pareto and exponential implementations of the algorithm, for the Cabspotting trace (Fig.

A.11). Regarding the ADD metric for the Expedited class, though, it can be noticed that both the Pareto implementation and the higher order approximations (i.e., second order and lower bound, section 5.2) based on the exponential distribution manage to capture the heterogeneous nature of the trace better and achieve lower ADD than the first order exponential approximation. This can be justified by the average number of copies evolution graph (Fig. A.13), where it is shown that, with the Pareto and the higher order exponentials, more copies are distiributed at the beginning of the bundles TTL. Thus, we confirm the more accurate and conservative nature of Pareto based predictions that we were expecting, comparing to the first order exponential ones, based on the results of sections A.2.2 and A.3. However, no difference is observed, in practice, regarding the comparison between the Pareto and the higher order exponentials.

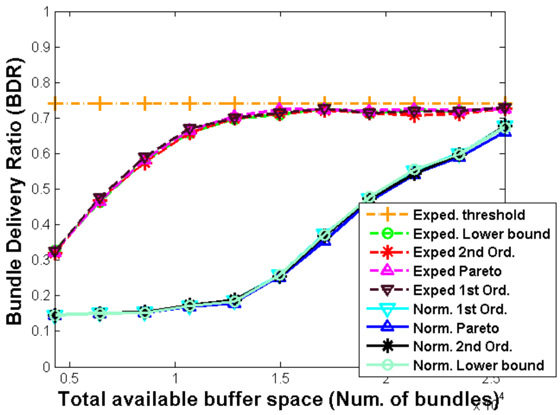

Figure A.11: Cabspotting: BDR Optimization, comparison of Pareto approx. with exponential based ones

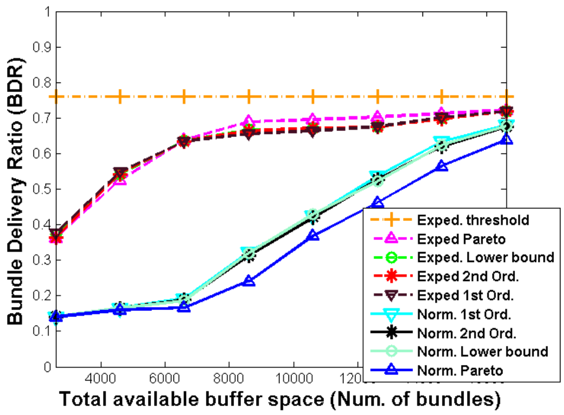

Figure A.12: SLAW: BDR Optimization, comparison of Pareto approx. with exponential based ones

On the contrary, in the case of the SLAW trace, it can be observed that the Expedited class achieves slightly higher BDR performance with the Pareto implementation and stabilizes closer to the desired QoS threshold, comparing to the three exponential ones (Fig. A.12). This, however, comes with the trade-off of lower performance for the normal class comparing to the family of exponential implementations. It can also be noticed that the performance difference with respect to the Normal class between the two distributions is higher than the respective difference with respect to the Expedited class. As discussed in section 5.3.3, this can be explained considering that a same amount of complementary resources leads to higher performance gains when it is given to bundles with low number of copies (i.e., Normal class), than when it is given to bundles with an already high number of copies (i.e. Expedited class). Thus, it can be deduced that with the generalized Pareto implementation, our policy becomes more conservative regarding the required resources to capture the QoS requirement of the Expedited class, providing less resources to Normal class bundles. The more conservative character of the Pareto-based predictions for the SLAW trace can also be verified by inspecting figure A.15. There, it is obvious that the Pareto implementation gives more copies to Expedited class bundles at the beginning of their TTL, comparing to the Exponential implementations. Also, it can be noticed that there is no essential difference on the way the copies are distributed over time between the three exponential based implementations. This can be justified by the fact that

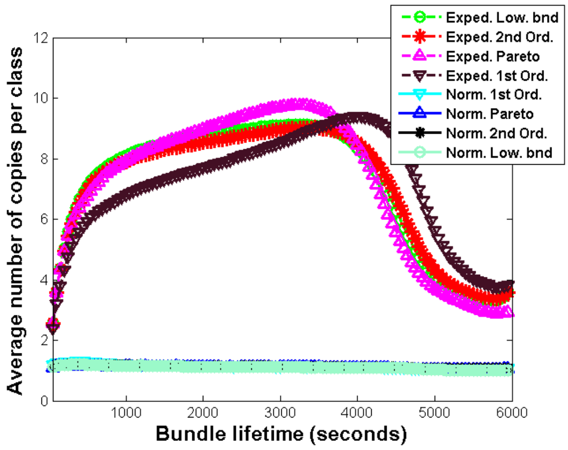

Figure A.13: Cabspotting: BDR Opt. Copies over time, comparison of Pareto approx. with exponential based ones

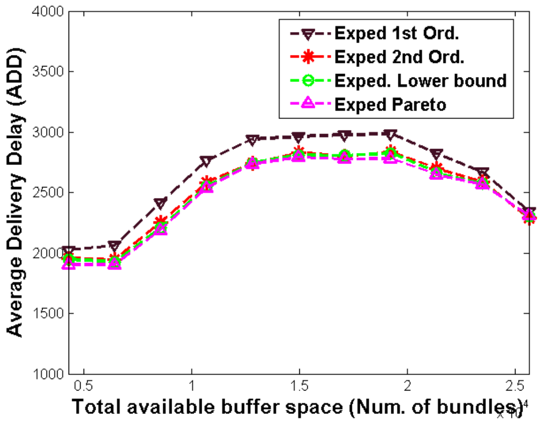

Figure A.14: Cabspotting: BDR opt. ADD metric, comparison of Pareto approx. with exponential based ones

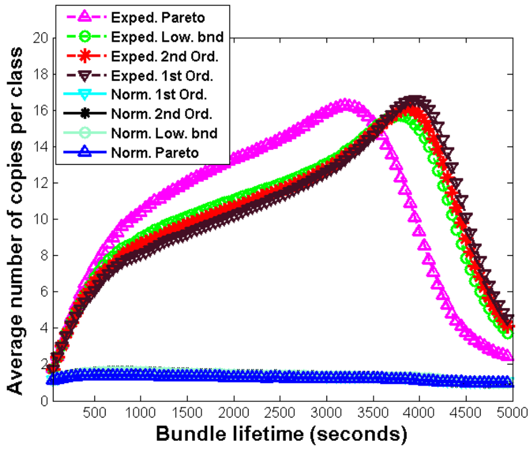

Figure A.15: SLAW: BDR Opt. Copies over time, comparison of Pareto approx. with exponential based ones

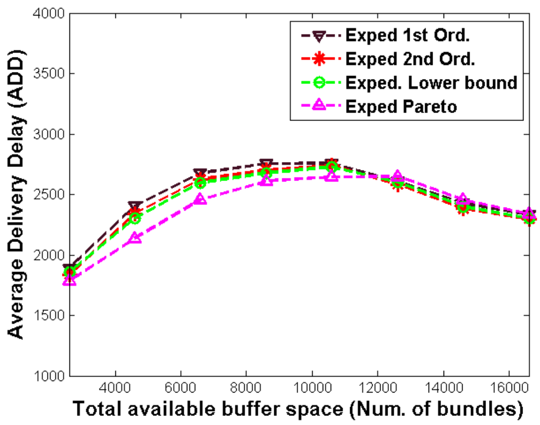

Figure A.16: SLAW: BDR opt., ADD metric, comparison of Pareto approx. with exponential based ones

the variance in the pairwise meeting rates is relatively small comparing to the value of mean meeting rate for the SLAW trace (Table A.1), as opposed to the other real mobility traces we have examined (i.e., Cabspotting and Infocom traces). As a result of this more homogeneous nature of contact rates in the SLAW trace, the more conservative approximations based on the exponential distribution for the BDR optimization problem do not make any significant difference in terms of the resources distribution for the Expedited class, in practice. The more conservative predictions with the Pareto distribution leads to constantly lower ADD for the Expedited class (figure A.16), comparing to the three exponential implementations, for a wide range of buffer values.

Overall, we can conclude that the Pareto-based implementation outperforms the first order exponential, for both mobility traces. When it comes to the comparison with the more conservative exponential approximations, no performance difference is observed with respect to

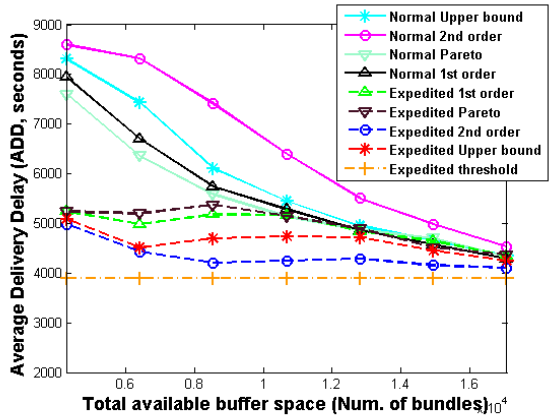

Figure A.17: Cabspotting: ADD Optimization, ADD metric: comparison of Pareto approx. with exponential based ones

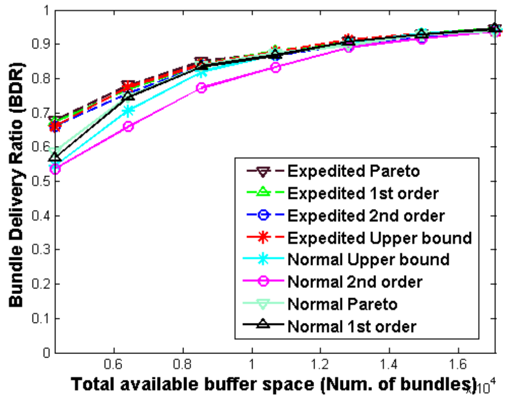

Figure A.18: Cabspotting: ADD Optimization, BDR metric: comparison of Pareto approx. with exponential based ones

the Cabspotting trace. However, for the SLAW trace, the Pareto implementation manages to capture the intended performance better, as it reaches to higher performance for the Expedited class, for both metrics of interest. This however comes with the cost of significantly lower performance with respect of the Normal class (Fig. A.12).

A.4.1.2 ADD Optimization

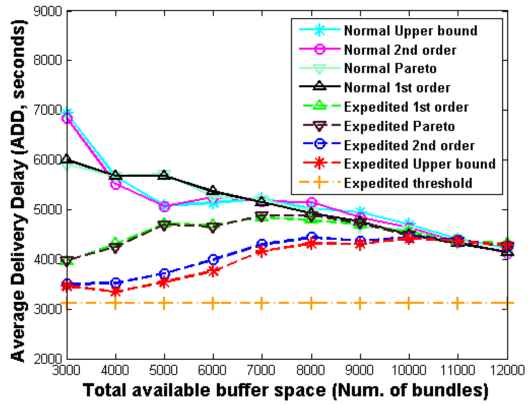

Figure A.19: SLAW: ADD Optimization, ADD metric: comparison of Pareto approx. with exponential based ones

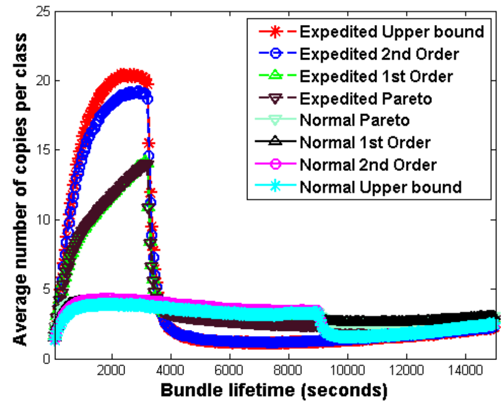

Figure A.20: SLAW: ADD Optimization, Copies evolution over time: comparison of Pareto approx. with exponential based ones

Let's turn our attention now to the performance of the Pareto implementation with respect to the ADD optimization problem (Fig. A.17 - A.21). Focusing on the Cabspotting trace, first, it can be noticed that the per class performance of the Pareto implementation doesn't provide any benefits comparing to any of the other exponential based approximations, achieving similar performance to the first order approximation and worse than the higher order ones

(Fig. A.17). This can be explained considering that, similarly to the first order exponential, the Pareto approximation depends only on a mean pairwise inter-meeting time value, which is extracted from the respective distribution describing a specific mobility trace (Eq. A.7). Thus, although the generalized Pareto model describes the overall distribution more accurately than the exponential (Fig. A.4), this cannot be actually captured in this optimization problem, as opposed to the case of BDR optimization. Indeed, as shown previously, in the latter the dependency of the optimized metric (i.e., delivery probability) from the remaining TTL of each bundle i, R_i, allowed to exploit the higher accuracy of the Pareto predictions and achieve better performance than the first order exponential ones.

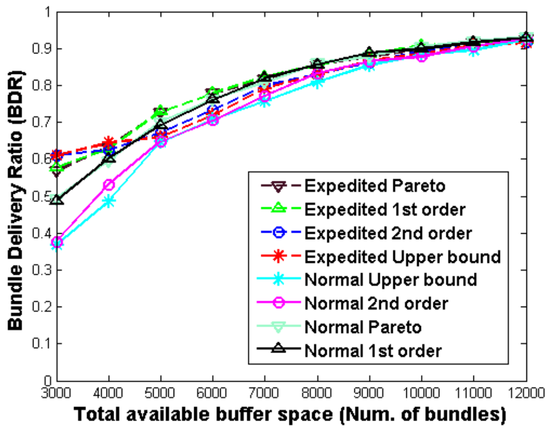

Figure A.21: SLAW: ADD Optimization, BDR metric: comparison of Pareto approx. with exponential based ones

For the case of the SLAW trace, once again the per QoS class performance in terms of ADD is the same between the first order exponential and the Pareto implementations (Fig A.19). The second order and upper bound exponential approximations outperform the other two, by making more conservative predictions and reaching closer to the average desired threshold of the Expedited class. This can once again be verified by inspecting figure A.20, where it is obvious that the higher order exponential approximations distribute more copies to the Expedited class. Thus, it is observed that, in contrast to the BDR optimization case for the SLAW trace and despite the relatively lower variance of pairwise meeting rates comparing to the other traces we have analyzed, the higher order exponential approximations actually make some improvement difference here.

With respect to the Pareto based implementation, however, it can be verified through the SLAW trace as well, that it does not offer any performance benefit comparing to the other three approaches in the context of the ADD optimization problem, for the reasons explained previously.

A.5 Conclusions

In this chapter, our aim was to examine the performance of our prioritization scheme, when the generalized Pareto function is used to model the pairwise inter-meeting times distribution, instead of the exponential. To this end, we first came out with a framework to extract the appropriate type of residual inter-meeting times distribution from each mobility trace, which considers all the pairwise statistics as equally important. In this context, we have shown that the generalized Pareto function can capture the extracted type of distribution better than the first order exponential, for the majority of mobility traces, in hand. This leads to the Pareto predictions outperforming the first order exponential ones for simple resource and QoS unconstrained scenarios based on different mobility traces.

Then, we focused on the actual QoS prioritization problems of BDR maximization and ADD minimization, by launching a Pareto based implementation of the policy described in chapter 4. There, it was interesting to observe that the comparative results of the distinct approximation methods (i.e., family of exponentials vs generalized Pareto) had some qualitative differences with respect to the optimization problem and the mobility trace, in hand. Thus, for the BDR optimization case, the Pareto implementation outperformed the first order exponential for all the evaluated mobility scenarios. Regarding the comparison with the higher order exponentials, though, the Pareto implementation achieved similar performance with them for the Cabspotting trace, but slighlty closer to the intended performance for the case of the SLAW trace. This lead us to the conclusion that the generalized Pareto distribution was the most suitable approximation method for the specific trace, in the context of BDR optimization, due to the latter's relatively short pairwise meeting rate variance and strong power-law component. Focusing on the ADD optimization problem, though, we observed that the first order Pareto model did not bring any performance benefits for the evaluated scenarios, comparing to any of the rest approximations.

Overall, we can conclude that the Pareto based prediction model, although performed better than the first order exponential for all the evaluation scenarios with respect to BDR optimization, it performed equally or worse than the higher order exponentials for the majority of evaluation scenarios, with respect to both optimization problems. To this end, we tend to consider the more conservative exponential approximation methods, as more appropriate for the purpose of our prioritization scheme, due also to the easier and more straightforward extraction of the necessary inter-meeting time distribution parameters of the mobility traces (i.e., mean rate and variance), comparing to the Pareto case.

Appendix B

An IPv6 Architecture for Cloud-to-Vehicle Smart Mobility Services over Heterogeneous Vehicular Networks

B.1 Introduction

Connected vehicles in smart cities of the future are envisioned to provide passengers with a wide range of services to facilitate their traveling experience. Such services will range from safety warnings and traffic information to mobility and comfort (infotainment) applications. In order to be able to host such services reliably, two basic requirements need to be fulfilled: *continuous connectivity between vehicles (V2V) and the Cloud (V2C)* and *efficient data load management*. To satisfy those two requirements, it is necessary to exploit connectivity through Heterogeneous means (e.g. big cells for high connectivity vs small cells for high throughput, or Cellular based 3G/LTE vs DSRC access). Moreover, Infrastructure Nodes (IF-Nodes) connected to vehicles have to be utilized efficiently in order to avoid bottlenecks and to reduce handovers latency. IPv6 is a natural choice to this objective, as it can natively operate over heterogeneous technologies and has a proven record of efficient Internet traffic flow management.

Whereas some cloud-based services could generate intensive traffic streams (CCTV (Closed circuit Television), major software updates, voice), most of the cloud services for smart mobility will bear resemblance with MTC (Machine-type communications): low individual volume, periodic traffic, large *aggregated* traffic to a large amount of vehicles. The two major differences from traditional MTC are first that such cloud-based services will generate *downlink traffic* rather than uplink, and second that these services will have a limited geographical scope. The connectionless character of such services sets the ground for creating flexible and transparent traffic offloading mechanisms among the IF-Nodes which can provide Internet access to the vehicles.

In this context, IPv6 mobility management solutions through Heterogeneous Networks need to be adapted. First, the convergence of DSRC and cellular technologies require an adapted IPv6 addressing scheme. Unlike cellular technologies, vehicles may be simultaneously connected to several Roadside Units (RSUs). Considering RSUs acting as IPv6 routers [97] creates the necessity of concurrently handling multiple vehicles IPv6 addresses (associated with multiple technology interfaces) and makes hierarchical IPv6mobility management (e.g. [98], [99]) solutions inefficient.

Second, hierarchical IPv6 mobility management functions are designed to optimize handover for connections that need to remain active for long intervals. Multipath TCP (MP-TCP) [100] is another approach to manage handovers and handle traffic offloading among its active flows efficiently. In [101] and [102], MP-TCP is proposed for vertical handovers among distinct technology interfaces. Yet, both hierarchical IPv6 Mobility Management and MP-TCP have been designed for cases when the active connection is long enough to justify handling handovers or maintaining multiple paths to the destination host. This does not hold for POI (MTC-like) traffic, where we are rather dealing with small bursts, as mentioned before. As a result, applying these approaches for POI traffic implies generating redundant overhead and signaling delay for short effective communication time. On the other hand, the fact that each vehicle can be accessible through multiple IF-Nodes at the same time broadens the capabilities for optimal per-packet dynamic traffic offloading among the IF-Nodes. In this context, the availability of updated geo-based information related to the vehicles can be of significant importance for this process.

Flat IPv6 architectures (e.g. Distributed Mobility Management (DMM) [103], Proxy DMM [104]) aim to overcome the weaknesses of architectures organized in multiple routing hierarchical levels (i.e. signaling overhead with centralized mobility anchor, non-optimal routing), by increasing direct communication among peers residing in the same geographical area [105]. However, such architectures are optimized for traffic initiated by the mobile node. Cloud-based services initiate traffic from the cloud, and the absence of a centralized *Home Agent* makes it difficult for cloud services to efficiently locate the target MN.

In this chapter, we propose an extension to flat-based IPv6 mobility management architectures to be adapted to cloud-initiated services over heterogeneous IF-nodes. We first formulate the specifications of a Point-of-Interest (PoI) application, periodically transmitting personalized information (e.g. public transportation info, car sharing, advertising and booking parking places and/or Electronic Vehicle charging spots etc.) to target vehicles. We then describe the new functions required to handle this application, namely LIMME *Location and Infrastructure Mobility Management Entity*, residing at the network Edge and consisting of three blocks: a Geographic Mobility Management (GMM) module, a location database mapping the MN IDs to the last known geographic locations and IPv6 address. Finally, an *IF-Node selector* to provide traffic offloading and optimize the communication capacity. As proof-of-concept, we implemented the required modules and extensions on the iTETRIS simulation platform [28] and illustrated their role in improved IPv6 mobility management over heterogeneous vehicular technologies.

The rest of the chapter is organized as follows. In section 2, we show the logic of our Point of Interest Application. In section 3, we discuss the requirements of the service. In section 4, we present our Network Architecture and its benefits. In section 5 we introduce the iTETRIS Architecture and the required extensions in Network Simulator (NS-3). In section 6, we show our proof-of-concept results on a basic scenario and in section 7 we highlight the conclusions of this work.

B.2 The Point Of Interest Cloud Application

The Point of Interest Application aims at providing an example of regional infotainment services which are offered to the set of vehicles entering the given region. The driver has the option to accept or decline the offered service and, once he accepts it, he starts receiving the data of this service periodically until he gets out of the region or he decides to stop the service.

This simple functionality is based on the exchange of three different types of messages between the Vehicles and the Infrastructure nodes (Wireless Access in Vehicular Environments (WAVE) RSUs or Cellular Base Stations):

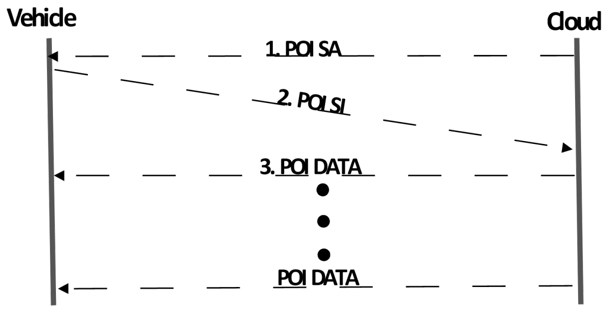

Figure B.1: POI App messages exchange

- **POI Service Advertisements (POI SA):** As the term indicates these messages are broadcasted periodically by the Infrastructure nodes (IF-Nodes) which cover a target geographical area, in order to advertise the offered service to the passing vehicles.
- **POI Service Interest (POI SI):** These Unicast messages are generated from the vehicles as a result of the reception of the Service Advertisements in order to subscribe to the specific service.
- **POI Service Data (POI DATA):** After receiving the Service Interest from a given vehicle, the Cloud starts generating Unicast Data messages for the subscribed vehicle periodically.

B.3 POI service Requirements

POI services constitute a family of applications which balance between their usefulness in a local scope and their need for Internet connectivity. We will now introduce the requirements of our envisioned service (Fig.B.2).

1. **On-demand service provision**. The vehicle will trigger the transfer of periodic service data by indicating to the Cloud its interest for the service.
2. **Periodic but independent data transmissions from the Cloud**. No need to provide support for continuous sessions and maintain active routes open is needed and, as a result, no need for traditional means of handovers. Thus, the traffic type resembles M2M but it comes from the Cloud not from the connected vehicles.

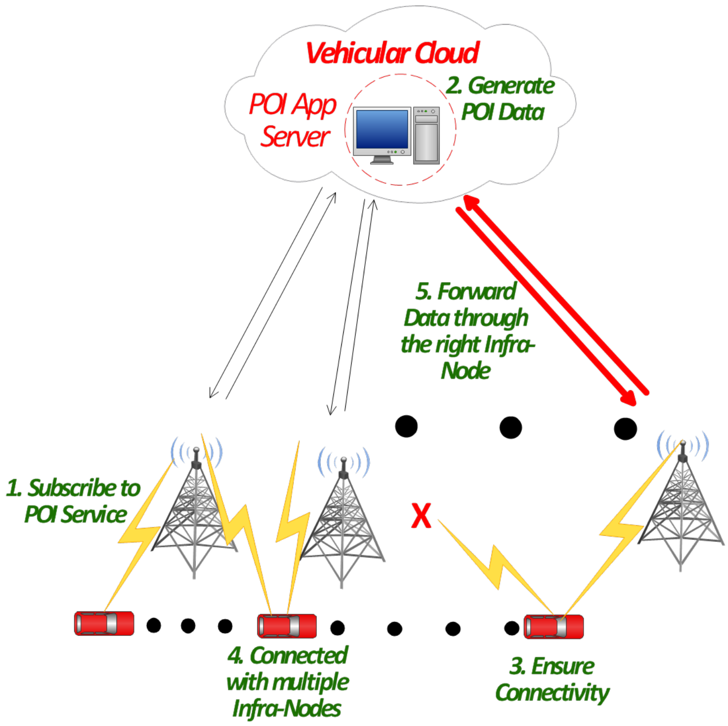

Figure B.2: POI service requirements

3. Our basic performance criterion is to **maintain connectivity** through some IF-Node during any data transfer.

4. **Vehicles can be connected to multiple IF-Nodes** even through the same technology interface (DSRC allow this, as the mobile nodes do not have to be part of a Basic Service Set (BSS)). As a result, vehicles can have multiple IPv6 addresses per interface at the same time.

5. Cloud to Vehicle (C2V) **traffic should be forwarded through the most appropriate IF-Node**, in terms of optimizing traffic load management.

6. **Localization services should converge with IPv6 operations** in order to provide IPv6 addressing and scalable Mobility Management services.

Concerning IF-Node selection at the downlink, it should be based on Lightweight local (channel load, Channel Quality Indicator (CQI)) or geographic (location, distance) metrics at IF-Nodes, rather than on testing single links quality between the IF-Nodes and the attached vehicles (changing very frequently with mobility, requires signaling overhead). Finally IPv6 Mobility Management should avoid the inefficiencies of centralized approaches such as MIPv6 [99] and Proxy MIPv6 [98] (e.g. long routing paths, signaling overhead, scalability issues etc.). The use of independent sessions implies that a **Distributed Mobility Management (DMM)-like approach** is more appropriate [103], [105].

B.4 Network Architecture

Based on the aforementioned requirements, we need a mechanism to select the best IF-Node for the downstream traffic from the Cloud, as well as a function which complements IPv6 addressing with location services in order to ensure IPv6 connectivity from the Cloud. In this context, our Architecture introduces the Location Infrastructure and Mobility Management Entity (LIMME), a new block which is located at the Edge of the Network, directly connected with the IF-Nodes of a given region where the POI service is offered. As the name indicates, this entity performs three basic functions namely, **Location Management**, **IF-Node selection** for data forwarding and **IPv6 Geographic Mobility Management (GMM)**. A high level view of these functions and the way they interact to handle cloud-originating IPv6 traffic efficiently is shown in Fig. B.3. IPv6 GMM provides the Location Management block with updated **Geographic Address Set (GAS)** information for the subscribed vehicles. This information is then queried by the IF-Node selector, based on the destination vehicle's IPv6 Home Address (HoA) [1] and utilized on the data plane to route IPv6 traffic through the appropriate IF-Node. We will now describe the roles of the separate LIMME functions in more detail.

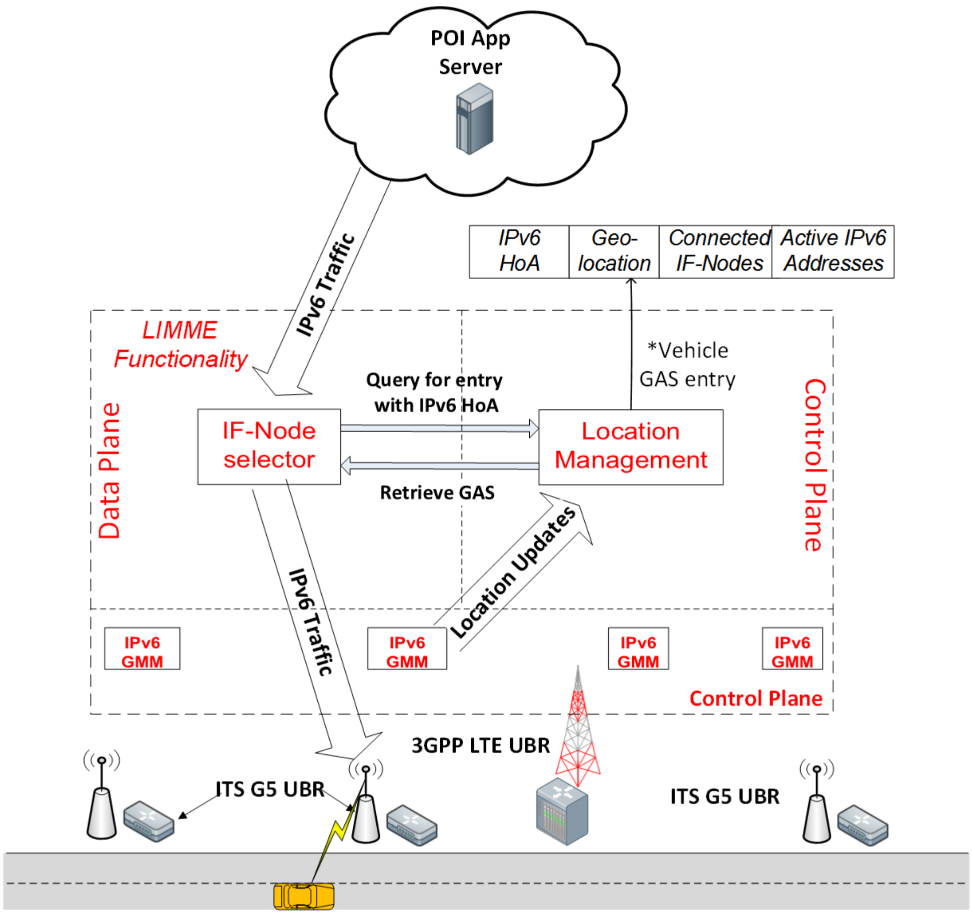

Figure B.3: POI App supporting Architecture

[1] By IPv6 HoA we indicate the initial IPv6 address with which the vehicle is subscribed to the service.

B.4.1 Location Management

The role of the Location Management function is to ensure that every subscribed vehicle can be tracked and accessed by LIMME through the IF-Node to which it is connected.

For DSRC, this functionality can be based on periodic Beacons broadcast by the vehicles via their C2C stack [2], where they report their geo-location to their neighbors (other vehicles and RSUs). Upon subscription to the POI Application, vehicles Location Updates (LUs) are periodically communicated to LIMME through their neighboring RSUs (see (c) and (g) in Fig. B.4). LIMME's Location Management block then updates its database with the vehicle's most recent GAS information. As shown in Fig. B.3, GAS includes information about: the vehicle's location, its IPv6 HoA, its neighbor IF-Nodes (location, IPv6 address), as well as its active IPv6 addresses. As we will show next, these addresses are obtained through the interaction with the IPv6 GMM block.

For Cellular coverage, location management can be based on Base Stations reporting the attached vehicles which are registered to the service. Although precise geographical information about the vehicle cannot be extracted in this case, this report is adequate to indicate to LIMME that the subscribed vehicle is reachable through cellular infrastructure.

As depicted in Fig. B.3, Management plane traffic (i.e. Location Updates) is in the uplink between the IF-Nodes and the Location Management. Given that such information is useful only for vehicles which have subscribed to the given regional service, the transferred information will concern those vehicles only, thus limiting the traffic on the uplink. This approach has also a low overhead, as the limited geographical scope of the service implies that only a limited set of IF-Nodes is expected to update MN locations at a time.

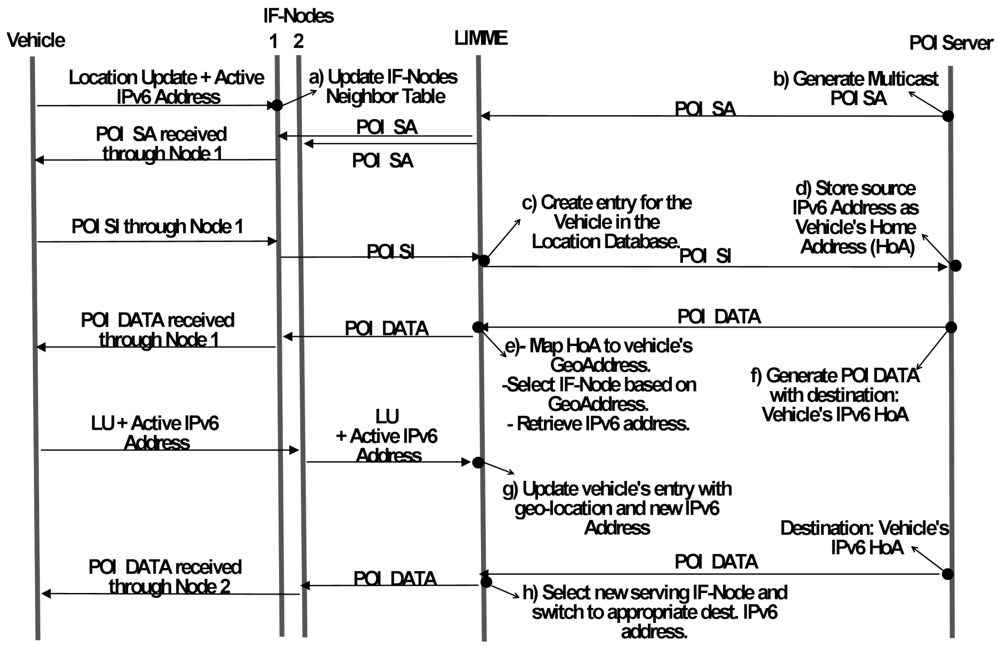

Figure B.4: LIMME Functionality

[2]C2C stack refers to the ETSI Geo-networking non-IP stack.

B.4.2 IPv6 Geographic Mobility Management (GMM)

The role of the IPv6 GMM function of LIMME is to extend the access to the vehicle nodes, from the Cloud at any point in time.

In our Architecture, every RSU or Cellular Base station acts as an IPv6 router which advertises its prefix and allows each passing by vehicle to autoconfigure a new global IPv6 address [106]. In that sense the Architecture is flat and it bears similarities with DMM [103]. In order to ensure connectivity with the Cloud, we extend our Location management function with IPv6 addressing support capabilities.

For DSRC this task is done in two steps. The first step concerns the Vehicle to IF-Node (V2I) plane and it dictates that the Location Management Beacons should be enriched with IPv6 addressing information. In this way, every IF-Node receiving Beacons can obtain IPv6 reachability information about the neighbor vehicles (see (a) from Fig. B.4).

To this direction, we suggest an extension which is based on the operation of the C2C stack (Geonetworking). Every node keeps a location table where it stores the information provided by the Location Management beacons about the other nodes. We suggest the enhancement of this information with the IPv6 addresse(s), attributed to the DSRC interface of each transmitting vehicle, as shown in Fig. B.5.

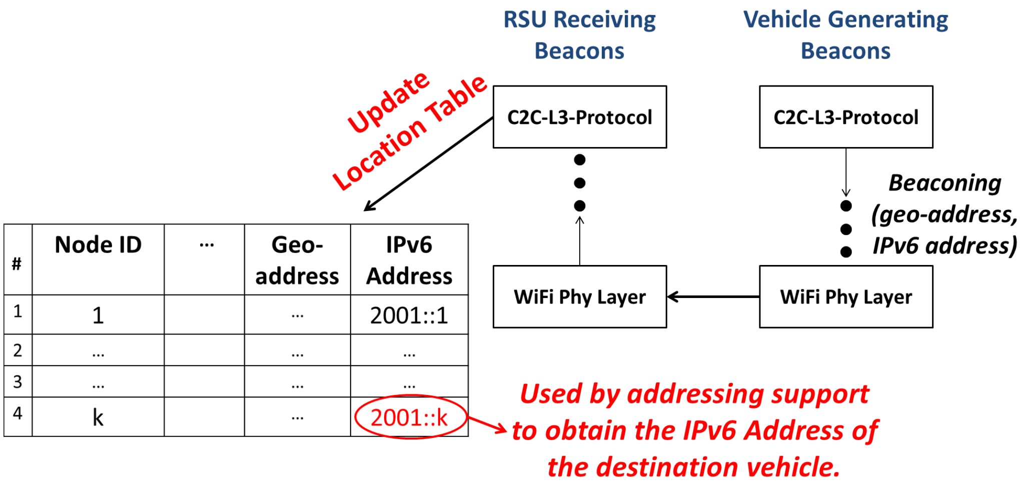

Figure B.5: IPv6 address retrieval through the C2C stack

The second step refers to the Infrastructure-to-LIMME plane and it is common among DSRC and cellular IF-Nodes. Based on this, every GAS update in LIMME's Location Management block includes IPv6 addressing information about the vehicle (see (g) from Fig. B.4).

B.4.3 IF-Node selection

The IF-Node selection block is responsible for choosing the most appropriate IF-Node for a given POI Data transmission.

Figure B.6 depicts the flow diagram for IF-Node selection. GAS information provided through the interaction with the Location Management Block specifies the set of candidate IF-Nodes to route the traffic through at each transmission. The Offloading Manager then picks

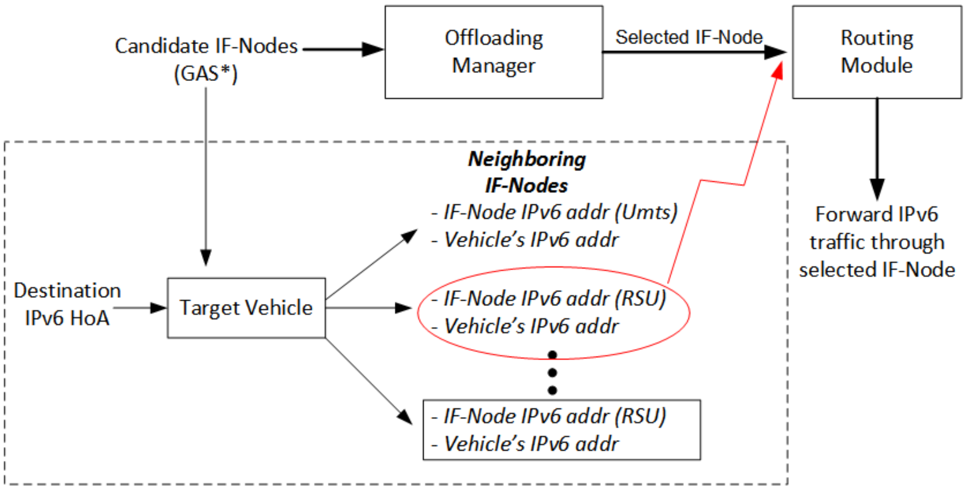

Figure B.6: IF-Node selection functionality

the best candidate based on geographical context information retrieved through GAS (see (e) and (h) from Fig. B.4), as well as other information provided by the IF-Nodes of the Application region. Such information can consider the minimum distance of the vehicle from the IF-Nodes of the area, as well as metrics extracted by the IF-Nodes (e.g. channel load for DSRC, Channel Quality Indicators (CQI) for cellular etc.), without the need of any additional management plane signaling which could create overhead.

The role of the routing module is to forward the traffic through the selected IF-Node, by mapping the vehicle's IPv6 HoA to the one with which it is accessible through the selected IF-Node (see (e) and (f) from Fig. B.4).

B.5 Implementation

We extended iTETRIS vehicular simulation platform to support our suggested Architecture and our POI Application logic. As proof of concept, we implemented the LIMME functionality and the POI Application on a synthetic scenario.

B.5.1 iTETRIS Simulator and Contributions

iTETRIS [28] is a platform for vehicular communication simulations which permits to define large scale realistic road and network traffic scenarios and simulate them through the integrated Network Simulator NS-3 [27], the road traffic simulator: Simulation of Urban Mobility (SUMO) [107] and an application module. One of the main assets of iTETRIS is that it provides, through its control system (iCS), the capability of bidirectional interactions between its Network and Mobility Simulator. The application block specifies the logic of different smart mobility applications.

In Fig. B.7 we summarize the placement of our extensions within the iTETRIS simulator. In the Application block we implemented the POI Application logic discussed in B.2. Thus, the Application Block represents the Cloud plane for our simulations. In the NS-3 block, we implemented all the IPv6-related communication extensions needed to support our Architecture.

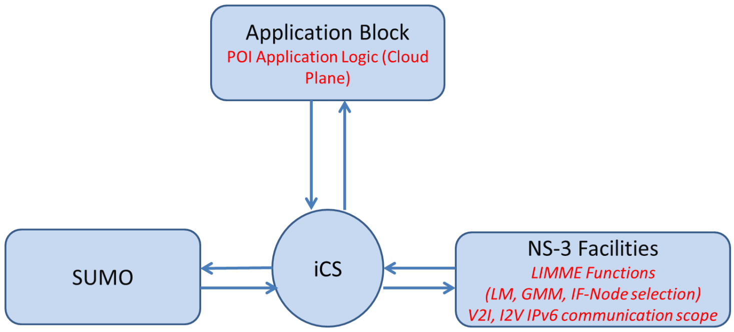

Figure B.7: Placement of Architecture and POIApp modules in iTETRIS

B.5.2 NS-3

iTETRIS uses an extension to the standard version of NS-3, which supports a socket interface with iCS, and includes an ETSI compliant ITS-G5A interface and an implementation of the geonetworking (C2C) stack, compliant with the ETSI Architecture [82]. Figure B.8 summarizes the ITS-related extensions, including the extended functionalities for LIMME.

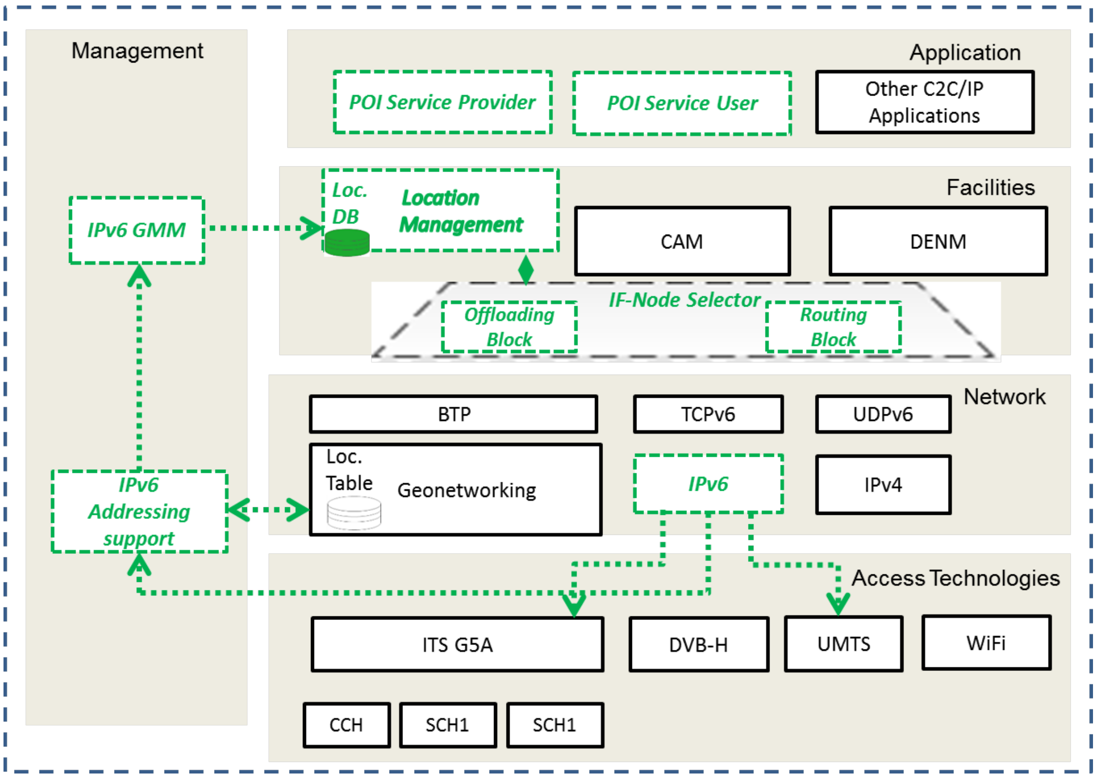

Figure B.8: NS-3 ITS extensions

B.5.2.1 Network block contributions

We extended the iTETRIS version of NS-3 with an IPv6 stack, which can ensure traffic offloading between cellular and DSRC communications, in a transparent way to the Cloud-based POI Application. In this context we had to extend the DSRC equipped nodes (i.e. vehicles and RSUs) with IPv6 addressing and routing capabilities, in the way that we described in section B.4.2.

B.5.2.2 Facilities and Management block contributions

These two blocks were extended to implement the functionality of the LIMME entity. Particularly, they are responsible for retrieving GAS information about the target vehicles, selecting the most appropriate IF-node for each POI DATA transmission and providing IPv6 reachability towards the target vehicle through the selected IF-Node. The extensions pertain also to IPv6 interoperability with the Management functions of different technology interfaces (i.e. DSRC, Universal Mobile Telecommunications System (UMTS)).

B.6 Simulation Results

For our simulations we launch an urban scenario where vehicles equipped with both UMTS and DSRC interfaces cross a straight road 1km long covered by 1 UMTS BS, which complements the coverage of 2 RSUs that provide only partial coverage in the area (Fig. B.9). In this region the POI service is available through the residing IF-Nodes and for every vehicle that is subscribed to the service, autonomous POI Data are generated from the Cloud.

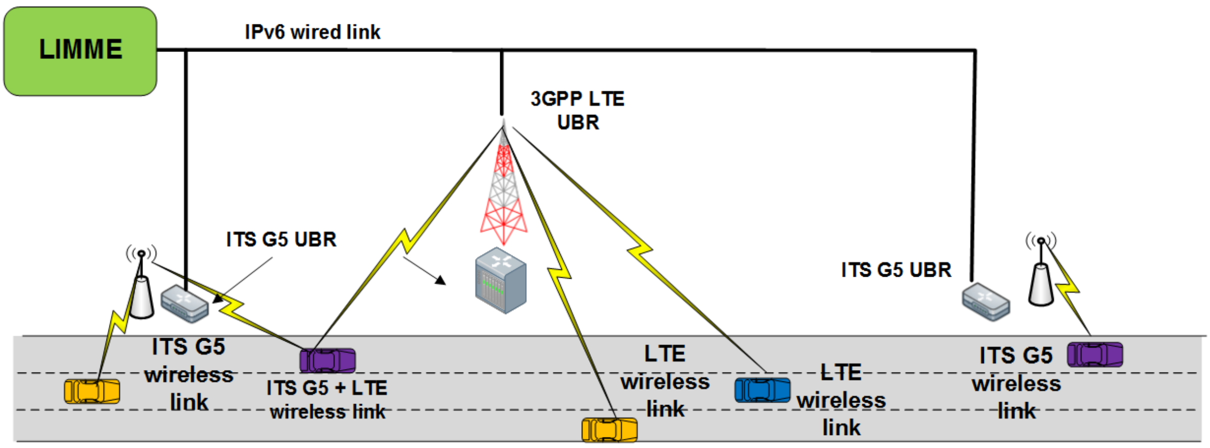

Figure B.9: Heterogeneous scenario

The selection among different technologies and IF-Nodes to forward the C2V traffic through is done by the LIMME, based currently on the following algorithm:

Algorithm 1 IF-NODE SELECTION

```
Retrieve set of neigbor IF-Nodes
if (#of IF-nodes == 1) then
    Select this node
else if (#of IF-nodes > 1) then
    if (All IF-Nodes are of the same techno) then
        Select the one with minimum Euclidean distance from the vehicle
    else
        Select the RSU with minimum Euclidean distance from the vehicle
    end if
end if
```

The capacity of cellular networks strongly depends on the number of connected nodes. In this context, the aim of traffic offloading is to maintain this capacity functional, regardless of the node density. In the following, the impact of vehicle density is evaluated. Our performance metrics are: the POI DATA Packet Delivery Ratio (PDR) and the Average Delivery Delay (ADD).

We introduced vehicle fleets of 5, 10, 20 and 30 vehicles for an Heterogeneous (2 DSRC RSUs, 1 UMTS BS available) and an Homogeneous scenario (2 DSRC RSUs available). The set of the simulation parameters is summarized in table B.1.

Parameters	Values
Number of Vehicles	5,10,20,30
Number of UMTS BS	1
Number of RSUs	2
RSU Inter-Distance	350 m.
Packet Size	1000 Bytes
Number of UMTS BS	1
Vehicular Speed	20 m/sec
Packet Generation Rate	1 pack/sec/vehicle
Data Rate on ITS-G5	6 Mbps
POI Penetration	100 %
ITS G5 Penetration	100 %

Table B.1: Vehicular Density simulation parameters

The obtained results are depicted in Fig. B.10 and B.11. In terms of PDR, we can see the benefit of using Heterogeneity, as the performance for the respective scenario is constantly above the Homogeneous one. This is explained by the full communication coverage provided when the UMTS BS is present. However, we notice that there is significant performance degradation for the Heterogeneous case, as we increase the number of vehicles. This is due to the limited number of data flows that are allowed to run concurrently within a UMTS Base station. Particularly, as we increase the number of vehicles, we also increase the number of flows that should be supported from the UMTS BS. However, due to a limit on the amount of data flows that can be handled concurrently by the BS, only a portion of vehicles can receive messages through the UMTS interface. On the other hand, for the Homogeneous case the performance is stable for 10,20 and 30 vehicles, while quite improved for 5 vehicles.

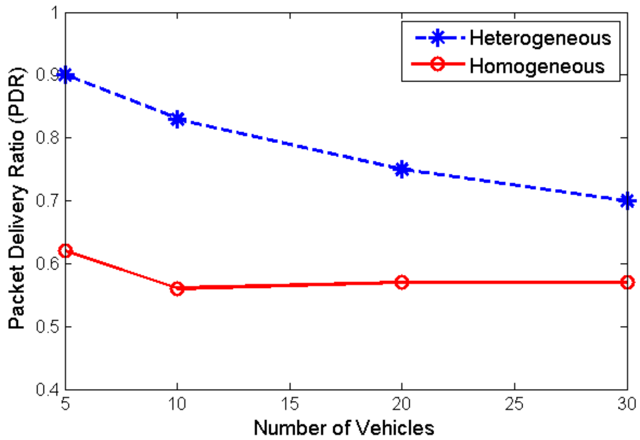

Figure B.10: Vehicle density vs overall Packet Delivery Ratio

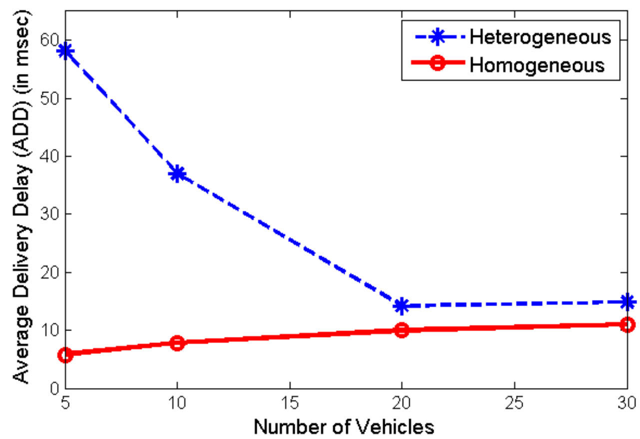

Figure B.11: Vehicular density vs Average Delivery Delay

In terms of ADD, we can once again view the impact of UMTS configuration restrictions in the curve of Heterogeneous scenario of Fig. B.11. Particularly, the percentage of total message receptions through UMTS interface is decreased as we increase the number of vehicles and the respective percentage for DSRC Interface is increased instead. As a result, this has an impact on the ADD which converges more to RSU Delivery Delay times (lower) than to UMTS ones (higher). For the Homogeneous scenario (pure DSRC) on the other hand we can view a normal behavior, where the Average Delivery Delay is slightly increased as we increase the vehicle density.

Despite the limits resulting from the restrictions in the configuration of UMTS, it is still obvious that increasing vehicular density reduces the performance of the PoI, in particular when using cellular networks. This further justifies why to rely on multiple RSUs to off-load exceeding traffic to vehicles, which would require investigations on metrics and triggers to decide when, and which flows should be off-loaded.

B.7 Conclusions

In this chapter, we proposed an IPv6 architecture for Cloud to Vehicle downstream services. This architecture is based on the convergence of geo-localization mechanisms with IPv6 addressing for ensuring reachability and traffic off-loading from the Cloud through optimal infrastructure node and technologies. From an IPv6 mobility management perspective, the IPv6 mobility management functions are located at the Network edge in order to better handle multiple IPv6 addresses identifying the same vehicle. But considering Cloud-initiated traffic, this architecture has been extended with a Geographic Mobility Management, where an entity called LIMME is in charge of identifying the optimal IF-Nodes to reach a particular vehicle. We implemented this Geographic Mobility Management architecture on the iTETRIS ITS simulation platform and illustrated its traffic offloading capabilities in a proof-of-concept.

Bibliography

[1] "Delay tolerant networking research group," http://www.dtnrg.org, 2016.

[2] J. Postel, "Transmission control protocol," *RFC 793*, September 1981.

[3] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The newreno modification to tcp's fast recovery algorithm," *RFC 6582*, April 2012.

[4] L. S. Brakmo, S. W. O'malley, and L. L. Peterson, "Tcp vegas: New techniques for congestion detection and avoidance," in *SIGCOMM*, 1994.

[5] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "Tcp westwood: End-to-end congestion control for wired/wireless networks," *Wireless Networks*, 2002.

[6] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost communication for rural internet kiosks using mechanical backhaul," in *Proc. ACM MobiCom*, 2006.

[7] I. Komnios and V. Tsaoussidis, "Carpool: Extending free internet access over dtn in urban environment," in *Proc. ACM MobiCom LCDNet '13 Workshop*, 2013.

[8] M. Allman, V. Paxson, and E. Blanton, "Tcp congestion control," *RFC 5681*, 2009.

[9] S. Floyd, "Congestion control principles," *RFC 2914*, 2000.

[10] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke University, Tech. Rep., 2000.

[11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM WDTN*, 2005.

[12] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," in *Proc. ACM ASPLOS*, 2002.

[13] "Routing protocols in vehicular delay tolerant networks: A comprehensive survey," *Computer Communications*, vol. 48, no. 0, 2014.

[14] Z. Lu and J. Fan, "Delay/disruption tolerant network and its application in military communications," in *Proc. IEEE ICCDA*, 2010.

[15] A. Balasubramanian, B. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *Proc. ACM SIGCOMM*, 2007.

[16] A. Krifa, C. Barakat, and T. Spyropoulos, “Message drop and scheduling in dtns: Theory and practice,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 9, Sept 2012.

[17] “Analysis of a QoS based routing strategy,” *MIDNET D3.3 deliverable*, May 2015.

[18] “Wireless communication architecture (land tactical): scenarios, requirements and operational view,” *NC3A Technical Note 1246*, 2007.

[19] R. Schäfer, K. Thiessenhusen, and P. Wagner, “A traffic information system by means of real-time floating-car data,” in *ITS world congress*, vol. 2, 2002.

[20] S. Ancona, R. Stanica, and M. Fiore, “Performance boundaries of massive floating car data offloading,” in *Proc. WONS*, 2014.

[21] A. Picu and T. Spyropoulos, “Dtn-meteo: Forecasting the performance of dtn protocols under heterogeneous mobility,” *IEEE/ACM Transactions on Networking*, 2014.

[22] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, “Power law and exponential decay of inter contact times between mobile devices,” in *Proc. ACM MobiCom*, 2007.

[23] H. Cai and D. Y. Eun, “Crossing over the bounded domain: From exponential to power-law inter-meeting time in manet,” in *Proc. ACM MobiCom*, 2007.

[24] V. Conan, J. Leguay, and T. Friedman, “Characterizing pairwise inter-contact patterns in delay tolerant networks,” in *Proc. ACM Autonomics*, 2007.

[25] W. jen Hsu and A. Helmy, “On nodal encounter patterns in wireless lan traces,” in *Proc. WiOpt*, 2006.

[26] A. Passarella and M. Conti, “Analysis of individual pair and aggregate intercontact times in heterogeneous opportunistic networks,” *IEEE Transactions on Mobile Computing*, 2013.

[27] http://www.nsnam.org/.

[28] M. Rondinone *et al.*, “ITETRIS: a modular simulation platform for the large scale evaluation of cooperative ITS applications,” *Simulation Modelling Practice and Theory, Elsevier, Volume 34*, 2013.

[29] V. Kerf *et al.*, “Delay-tolerant networking architecture,” *IETF RFC 4838*, April 2007.

[30] K. Scott and S. Burleigh, “Bundle protocol specification,” *IRTF RFC 5050*, August 2008.

[31] S. Burleigh, “Bundle protocol extended class of service (ecos),” *draft-irtf-dtnrg-ecos-05*, July 2013.

[32] I. F. Akyildiz, G. Morabito, and S. Palazzo, “Tcp-peach: A new congestion control scheme for satellite ip networks,” *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, 2001.

[33] C. Caini and R. Firrincieli, “Tcp hybla: a tcp enhancement for heterogeneous networks,” *International Journal of Satellite Communications and Networking, Vol:22*, 2004.

[34] D. Kandula, P. Pathak, and R. Dutta, “MF-TCP : Design and evaluation of tcp for message ferry delay tolerant networks,” in *Proc. ATNAC*, 2009.

[35] O. Akan, J. Fang, and I. Akyildiz, "Tp-planet: a reliable transport protocol for interplanetary internet," *IEEE Journal on Selected Areas in Communications*, 2004.

[36] M. Demmer, J. Ott, and S. Perreault, "Delay tolerant networking tcp convergence layer protocol," *IRTF draft-irtf-dtnrg-tcp-clayer-02*, 2008.

[37] I. Psaras, G. Papastergiou, V. Tsaoussidis, and N. Peccia, "Ds-tp: Deep-space transport protocol," in *Proc. IEEE Aerospace Conference*, 2008.

[38] B. Adamson, C. Bormann, M. Handley, and J. Macker, "Nack-oriented reliable multicast (norm) transport protocol," *IETF RFC 5740*, November 2009.

[39] C. Rigano, K. Scott, J. Bush, R. Edell, S. Parikh, R. Wade, and B. Adamson, "Mitigating naval network instabilities with disruption tolerant networking," in *Proc. IEEE Milicom*, 2008.

[40] S. Burleigh, M. Ramadas, and S. Farrel, "Licklider transmission protocol - motivation," *IRTF RFC 5325*, 2008.

[41] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider transmission protocol - specification," *IRTF RFC 5326*, 2008.

[42] L. Wood, W. Eddy, C. Smith, W. Ivancic, and C. Jackson, "A scalable data transfer protocol," *IETF draft-wood-tsvwg-saratoga-15*, April 2014.

[43] L. Wood, J. McKim, W. M. Eddy, W. Ivancic, and C. Jackson, "Using saratoga with a bundle agent as a convergence layer for delay- tolerant networking," *IETF draft-wood-dtnrg-saratoga-14*, April 2014.

[44] L. Wood, C. Smith, W. Eddy, W. Ivancic, and C. Jackson, "Taking saratoga from space-based ground sensors to ground-based space sensors," in *Proc. IEEE Aerospace Conference*, 2011.

[45] L. Wood, W. Eddy, and W. Ivansic, "Congestion control for the saratoga protocol," *IETF draft-wood-tsvwg-saratoga-congestion-control-09*, May 2016.

[46] H. Kruse, S. Jero, and S. Ostermann, "Datagram convergence layers for the delay- and disruption-tolerant networking (DTN) bundle protoco and licklider transmission protocol," *IRTF RFC 7122*, March 2014.

[47] S. Farrell and V. Cahill, "Ltp-t: A generic delay tolerant transport protocol," TCD Computer Science Technical Report TCD-CS-2005-69, Tech. Rep., 2005.

[48] D. Hua, X. Du, L. Cao, G. Xu, and Y. Qian, "A dtn congestion avoidance strategy based on path avoidance," in *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, vol. 1, May 2010, pp. V1–855–V1–860.

[49] E. Coe and C. Raghavendra, "Token based congestion control for dtns," in *Proc. IEEE Aerospace Conference*, 2010.

[50] M. Radenkovic and A. Grundy, "Congestion aware forwarding in delay tolerant and social opportunistic networks," in *Proc. IEEE WONS*, 2011.

[51] G. Zhang, J. Wang, and Y. Liu, "Congestion management in delay tolerant networks," in *Proc. ACM WICON*, 2008.

[52] T. Kathiravelu, N. Ranasinghe, and A. Pears, "An enhanced congestion aware adaptive routing protocol for opportunistic networks," in *Proc. IEEE ICIINFS*, 2011.

[53] D. G. Akestoridis, N. Papanikos, and E. Papapetrou, "Exploiting social preferences for congestion control in opportunistic networks," in *Proc. IEEE WiMob*, 2014.

[54] J. M. Pujol, A. L. Toledo, and P. Rodriguez, "Fair routing in delay tolerant networks," in *Proc.IEEE INFOCOM 2009*, 2009.

[55] J. Lakkakorpi, M. Pitkänen, and J. Ott, "Using buffer space advertisements to avoid congestion in mobile opportunistic dtns," in *Proc. ACM WWIC*, 2011.

[56] F. Bjurefors, P. Gunningberg, C. Rohner, and S. Tavakoli, "Congestion avoidance in a data-centric opportunistic network," in *Proc. ACM SIGCOMM ICN Workshop*, 2011.

[57] K. Wang, H. Guo, L. Shu, and B. Liu, "An improved congestion control algorithm based on social awareness in delay tolerant networks," in *Proc. IEEE ICC*, 2014.

[58] R. I. Ciobanu, C. Dobre, and V. Cristea, "Reducing congestion for routing algorithms in opportunistic networks with socially-aware node behavior prediction," in *Proc. IEEE AINA*, 2013.

[59] N. Kandhoul and S. K. Dhurandher, "Congestion control in social opportunistic networks," in *Proc. IEEE ICSPCT*, 2014.

[60] M. Radenkovic and A. Grundy, "Framework for utility driven congestion control in delay tolerant opportunistic networks," in *Proc. IEEE IWCMC*, 2011.

[61] N. Thompson, S. C. Nelson, M. Bakht, T. Abdelzaher, and R. Kravets, "Retiring replicants: Congestion control for intermittently-connected networks," in *Proc. IEEE INFOCOM*, 2010.

[62] Y.-K. Ip, W.-C. Lau, and O.-C. Yue, "Forwarding and replication strategies for DTN with resource constraints," in *Proc. IEEE VTC*, 2007.

[63] S. C. Lo and C. L. Lu, "A dynamic congestion control based routing for delay-tolerant networks," in *Proc. FSKD*, 2012.

[64] E. Wang, Y. Yang, J. Wu, and W. Liu, "A buffer management strategy on spray and wait routing protocol in dtns," in *Proc. IEEE ICPP*, 2015.

[65] A. Elwhishi, P. H. Ho, K. Naik, and B. Shihada, "A novel message scheduling framework for delay tolerant networks routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, 2013.

[66] K. Wei, S. Guo, D. Zeng, and K. Xu, "A multi-attribute decision making approach to congestion control in delay tolerant networks," in *IEEE ICC*, 2014.

[67] A. Lindgren, A. Doria, E. Davies, and S. Grasic, "Probabilistic routing based on history of messages in delay tolerant networks," *IRTF RFC 6693*, August 2012.

[68] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE INFOCOM*, 2006.

[69] R. D. S. Symington and K. Scott, "Delay-tolerant networking custodial multicast extensions," *IETF draft-symington-dtnrg-bundle-multicast-custodial-06*, August 2009.

[70] K. Srinivasan and P. Ramanathan, "Reliable multicasting in disruption tolerant networks," in *Proc. IEEE GLOBECOM*, 2010.

[71] K. A. Harras and K. C. Almeroth, "Transport layer issues in delay tolerant mobile networks," in *Proceedings of the 5th International IFIP-TC6 Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, ser. NETWORKING'06, 2006.

[72] P. Raveneau, R. Dhaou, E. Chaput, and A. L. Beylot, "Dtns back: Dtns broadcasting ack," in *Proc. IEEE GLOBECOM*, 2014.

[73] S. Kaveevivitchai and H. Esaki, "Independent dtns message deletion mechanism for multi-copy routing scheme," in *Proc. ACM AINTEC*, 2010.

[74] Y. An, J. Huang, H. Song, and J. Wang, "A congestion level based end-to-end acknowledgement mechanism for delay tolerant networks," in *Proc. IEEE GLOBECOM*, 2012.

[75] A. Ali, E. Altman, T. Chahed, M. Panda, and L. Sassatelli, "A new reliable transport scheme in delay tolerant networks based on acknowledgments and random linear coding," in *Proc. IEEE ITC*, 2011.

[76] A. Ali, M. Panda, T. Chahed, and E. Altman, "Improving the transport performance in delay tolerant networks by random linear network coding and global acknowledgments," *Elsevier Journal on Ad Hoc Networks*, vol. 11, no. 8, 2013.

[77] P. U. Tournoux, E. Lochin, J. Leguay, and J. Lacan, "Robust streaming in delay tolerant networks," in *Proc. IEEE ICC*, 2010.

[78] B. Lorenzo, S. Glisic, L. Galluccio, and Y. Fang, "Adaptive infection recovery schemes for multicast delay tolerant networks," in *Proc. IEEE GLOBECOM*, 2013.

[79] G. Sandulescu and S. Nadjm-Tehrani, "Opportunistic dtn routing with window-aware adaptive replication," in *Proc. ACM AINTEC*, 2008.

[80] V. Soares, F. Farahmand, and J. Rodrigues, "Traffic differentiation support in vehicular delay-tolerant networks," *Springer Journal on Telecommunication Systems*, vol. 48, no. 1-2, 2011.

[81] "Traffic management strategy for delay-tolerant networks," *Elsevier Journal on Network and Computer Applications*, vol. 35, no. 6, 2012.

[82] "ETSI TC ITS, Intelligent Transport Systems; Communications Architecture, Standard ETSI EN 320 665, v 1.1.1," 2010.

[83] "Evaluation of DTN routing strategies against military requirements," *MIDNET D3.4 deliverable*, September 2015.

[84] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*. Boston, MA: Springer US, 1996, pp. 153–181.

[85] "Midnet architectural model and common interface description," *MIDNET WP3 Architectural Annex*, May 2015.

[86] P. Matzakos and J. Haerri, "Dtn mechanisms and architectures applied in hybrid vehicular networks," *Research contract-Orange labs: D2.2 deliverable*, March 2015.

[87] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[88] C. Edwin K. P. and Z. Stanislaw H., *An Introduction to Optimization*. New Jersey, USA: John Wiley & Sons, 2013.

[89] P. Sermpezis and T. Spyropoulos, "Delay analysis of epidemic schemes in sparse and dense heterogeneous contact environments," Eurecom, Tech. Rep. RR-12-272, 2012.

[90] G. W. Oehlert, "A note on the delta method," *The American Statistician*, 1992.

[91] E. Cooch and G. White, "Program mark: a gentle introduction," *download at http://www. phidot. org/software/mark/docs/book*, 2006.

[92] S. P. Dokov and D. P. Morton, "Higher-order upper bounds on the expectation of a convex function," *Stochastic Programming EPrint Series*, 2002.

[93] A. Madansky, "Bounds on the expectation of a convex function of a multivariate random variable," *Ann. Math. Statist.*, 1959.

[94] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset epfl/mobility (v. 2009-02-24)," Downloaded from http://crawdad.org/epfl/mobility/20090224, 2009.

[95] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD dataset cambridge/haggle (v. 2006-09-15)," Downloaded from http://crawdad.org/cambridge/haggle/20060915/imote, 2006.

[96] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Slaw: A new mobility model for human walks," in *Proc. IEEE INFOCOM*, 2009.

[97] A. Petrescu, N. Benamar, and T. Leinmueller, "Transmission of ipv6 packets over ieee 802.11p networks," *IETF draft-petrescu-ipv6-over-80211p-02*, December 18 2014.

[98] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile Ipv6," *IETF RFC 5213*, August 2008.

[99] C. Perkins, D. Johnson, and J. Arkko, "Mobility support in Ipv6," *IETF RFC 6275*, July 2011.

[100] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," *IETF RFC 6824*, January 2013.

[101] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/wifi handover with multipath TCP," *Proc. ACM SIGCOMM workshop on Cellular Networks*, 2012.

[102] C. Raiciu, D. Niculescu, M. Bagnulo, and M. Handley, "Opportunistic mobility with multipath TCP," *Proc. ACM MobiArch*, 2011.

[103] C. Bernardos, A. de la Oliva, and F. Giust, "An IPv6 distributed client mobility management approach using existing mechanisms," *IETF draft-bernardos-dmm-cmip-01 work in progress*, August 2014.

[104] J. Lee and Y. Kim, "Pmipv6-based distributed mobility management," *IETF draft-jaehwoon-dmm-pmipv6-02 work in progress*, May 20 2014.

[105] H. Chan, D. Liu, P. Seite, H. Yokota, and J. Korhonen, "Requirements for distributed mobility management," *IETF draft-ietf-dmm-requirements-16 work in progress*, April 18 2014.

[106] S. Thomson, T. Narten, and T. Jinmei, "Ipv6 stateless address autoconfiguration," *IETF RFC 4862*, September 2007.

[107] http://sumo-sim.org/.