UNIVERSITÀ DEGLI STUDI DI SIENA 1240

Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione
e Scienze Matematiche

Doctor of Philosophy Dissertation in Information Engineering

Efficient M2M Communications via Random Access Satellite Channels

by
Felice Manlio Bacco

Supervisor: Prof. Giovanni Giambene
Co-Supervisor: Dr. Alberto Gotta

Siena, Italy
October 18, 2016

Contents

Abstract

According to data traffic forecast reports, the volume of data transported by Internet in 2020 will exceed the threshold of 2.0 zettabytes per year, generated by more than trillion of devices. Only a minor portion of the traffic will be generated by PCs, as commonly observed in the recent past. On the contrary, a large quota of Internet traffic is expected to be generated by TVs, tablets, smartphones, and Machine To Machine (M2M) devices. In particular, it has been highlighted that M2M traffic will experience a growth rate in the order of 60%. M2M refers to technologies that allow both wireless and wired systems to communicate with other devices of the same ability, sensors and actuators, without any human intervention.

M2M applications are largely diffused in several deployments and have pushed the scientific community to thoroughly investigate the network design implications. The large amount of traffic contributed by these applications will have an important impact on the design of future network architecture and on dimensioning the capacity of the telecommunication infrastructures. From this standpoint, a special note has to be reserved to the case of M2M services distributed via satellite, whose related industry is continuously increasing in size, which is the main focus of this work. More generally, the research problem can be stated as follows: a large and dense population of M2M devices exchanges short data bursts over a shared satellite medium; support for sporadic and unpredictable access activity and/or support to delay-critical applications is required. Random Access (RA) schemes for handling uncoordinated multiple access can nowadays compete with the throughput offered by typical coordinated techniques, making the former ones strongly attractive to support large populations of M2M terminals, while contemporary providing

immediate access to the channel, without any reservation delays that are typical of coordinated access schemes. Therefore, the focus of this Ph.D. Thesis is on analysing, analytically and empirically, the behaviour of the most common M2M protocol stacks on RA satellite links and on suggesting guidelines to improve the achievable performance level.

The first part of this Thesis presents a load control algorithm to maximize the throughput achievable by Return Link Satellite Terminals (RCSTs), focusing on the exploitation of linear code prior to transmission, in order to improve the transmission robustness at the cost of some capacity waste because of the redundancy. Furthermore, an innovative hybrid access protocol is designed, aiming at allowing the coexistence of M2M and non-M2M RCSTs in the same network.

The second part of this Thesis studies the performance of M2M protocol stacks via RA satellite channels, when a reliable data delivery is needed in channels suffering of erasures due to collisions. Two metrics are taken into account: the completion time, in presence of short data bursts, and the throughput, in presence of a sustained load. A cross-layer study is proposed, in order to characterize the interactions among application, transport and Media Access Control (MAC) layers of a M2M protocol stack sending data via a RA satellite channel.

Acknowledgements

La conclusione di questo dottorato segna un traguardo importante. Per il supporto ricevuto durante questi anni, sono tante le persone che voglio qui ringraziare. In primis la mia famiglia, che è sempre stata dalla mia parte in ogni scelta fatta, permettendomi di tracciare la mia personale strada, negli studi e nel lavoro. Voglio ringraziare anche Margherita, la mia ragazza, con me in tutti i giorni di questo dottorato, difficili e non, e che mi accompagna a questo traguardo.

Vorrei ringraziare anche tutti coloro che hanno contribuito scientificamente, oltre che umanamente, al presente lavoro: il professor Giovanni Giambene dell'Università di Siena, mio advisor; il capo del Wireless Network Laboratory (WNLAB) dell'Istituto di Scienza e Tecnologie dell'Informazione (ISTI) del CNR di Pisa, con cui da anni collaboro attivamente, la dott.ssa Erina Ferro; il ricercatore del WNLAB Alberto Gotta, mio co-tutor in questo lavoro, nonché amico. Ancora, vorrei ringraziare l'intero WNLAB per la opportunità di aver collaborato con tanti di questo gruppo in questi anni. Infine, un grande ringraziamento anche a Tomaso De Cola, ricercatore dell'Institut für Kommnication und Navigation del DLR, Oberpfaffenhofen-Wessling, Baviera, per la supervisione durante il periodo di ricerca presso l'istituto per cui lavora e per la collaborazione ancora oggi attiva.

List of Figures

List of Tables

Acronyms

Acronym	Meaning
3WHS	Three Way Handshake
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
ACK	Acknowledgement
ARQ	Automatic Repeat reQuest
BER	Bit Error Rate
BLER	Block Error Rate
BLR	Burst Loss Rate
CA	Congestion Avoidance
CAFR	Congestion Avoidance - Fast Retransmit / Fast Recovery
CGC	Complementary Ground Component
CoAP	Constrained Application Protocol
CoRE	Constrained RESTful Environments
CRDSA	Contention Resolution Diversity Slotted Aloha
CSA	Coded Slotted Aloha
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CWND	Congestion Window
DA	Dedicated Access
DAMA	Demand Assigned Multiple Access

Acronym	Meaning
DSA	Diversity Slotted Aloha
DVB-H	Digital Video Broadcasting - Handheld
DVB-RCS2	Digital Video Broadcasting - Return Channel via Satellite, Second Generation
DVB-SH	Digital Video Broadcasting - Satellite services to Handhelds
ESA	European Space Agency
ETSI	European Telecommunication Standard Institute
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FR	Fast Recovery/Fast Retransmit
GW	Gateway
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet Of Things
IP	Internet Protocol
IRSA	Irregular Repetition Slotted Aloha
LMS	Land Mobile Satellite
LoS	Line of Sight
M2M	Machine To Machine
MAC	Media Access Control
MPE-IFEC	Multi-Protocol Encapsulation - Inter-burst Forward Error Correction
MSS	Maximum Segment Size
NC	Network Coding
NCC	Network Control Center
PLR	Packet Loss Rate
PUB/SUB	Publish / Subscribe

Acronym	Meaning
QoS	Quality of Service
QPSK	Quadrature Phase-Shift Keying
RA	Random Access
RCST	Return Link Satellite Terminal
RFC	Request For Comment
RLNC	Random Linear Network Coding
ROHC	Robust Header Compression
RSU	Road-Side Unit
RTO	Retransmission Timeout
RTT	Round Trip Time
SA	Slotted Aloha
SCADA	Supervisory Control and Data Acquisition
SIC	Successive Interference Cancellation
SLR	Segment Loss Rate
SNIR	Signal to Interference plus Noise Ratio
SNR	Signal to Noise Ratio
SS	Slow Start
ST	Satellite Terminal
TCP	Transport Control Protocol
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
WF	WaveForm
WSN	Wireless Sensor Network

Chapter 1

Introduction

According to data traffic forecast reports, the volume of data transported by Internet in 2020 will exceed the threshold of 2.0 zettabytes per year, and only a minor portion of the traffic will be generated by PCs; on the contrary, a large quota of Internet traffic is expected to be generated by M2M devices. M2M refers to technologies that allow both wireless and wired systems to communicate with other devices of the same ability. The core characteristic of M2M communications is the absence of a direct human data consumer/producer.

In the last 15 years, M2M communications have yielded the way to Internet Of Things (IoT) communications: in fact, IoT adds several key features to M2M, such that it can be considered an evolution of the initial idea of M2M, market-driven. The keystone of the IoT paradigm is *interoperability*: closed protocol stacks moving towards an open standard solution. An open protocol stack represents the key to implement the vast world of services and applications that the market provides day by day, thanks to the ever-going technological innovation: for instance, in smart grid, environmental and energetic monitoring, smart transportations, and healthcare scenarios, which are only just few examples of the market fields that already largely benefit of the IoT paradigm and its innovative features. IoT/M2M applications are today very diffused and the large amount of traffic that these applications carry along has a significant impact on the design of future network architectures and on dimensioning the capacity of the telecommunication infrastructures. IoT/M2M means that an very large number of connected devices must be

supported: they sporadically access the network, exhibiting complex time-patterns. This traffic pattern is different from the one of web traffic, and challenges the existing network infrastructure and the protocol stacks in use. The enormous amount of data produced by these devices must be reliably delivered for the IoT applications to properly function, and the industrial and scientific communities are struggling to cope with that. Given the exponentially increasing IoT/M2M traffic rate flowing in our networks nowadays and the large number of services and applications already in place, this challenge must be met.

Both industrial and scientific communities have shown a great interest in the complex scenario of IoT/M2M services, because of the large number of issues as well as of opportunities it carries along. Both communities look forward to a larger use of the advantages that the satellites may offer: in fact, the fraction of the M2M traffic delivered via satellite is continuously increasing in size [1, 2]. In [3], the authors give an overview on the most outstanding characteristics of the IoT/M2M paradigm and on its impact on the existing commercial services, focusing on the enabling technologies and on the strong standardization need, in order to avoid an useless and excessive fragmentation of the ecosystem. In particular, reference [3] highlights the challenge of finding a generally valid traffic model, which would represent an important piece of the dimensioning puzzle of the telecommunication systems. The telecommunication systems must support the existing and upcoming IoT/M2M services and applications, which are largely described in several white papers and industrial reports: for instance, the white paper by Deloitte titled *Machine-to-Machine: Vision 2020 - Is India ready to seize a USD 4.5 trillion M2M opportunity?* [4] analyses the specific characteristics of the Indian market, and underlines the fact that the satellite operators are considered commercial partners of primary importance. This conclusion is shared by other market analyses and by already M2M services provided via satellite; for instance, in transportation, agriculture, mining, oil and gas sectors, which provide increasing market revenues[1]. The use of satellites in supporting IoT/M2M applications has lead to the definition of the so-called

[1]The Northern Sky Research (NSR) company publicates detailed and up-to-date industrial reports about IoT/M2M services via satellite, as for instance in: `http://www.nsr.com/upload/research_reports/NSR_M2M6_Brief.pdf`

Internet of Remote Things (IoRT) in [5]; this scientific paper describes the different parts that compose the scenario of IoT/M2M services via satellite, highlighting several advantages: for instance, the *broadcast* nature that natively supports group services, quite common in IoT scenarios; the *ubiquitous* coverage, which cannot be guaranteed in rural areas by terrestrial services, for cost reasons; the use as *backup* backbones, which can prove vital in case of a malfunctioning of the terrestrial networks. From this standpoint, a special note has to be reserved to the case of IoT/M2M services distributed via RA satellite channels. The European Space Agency (ESA), among other main industrial and scientific actors, has shown a large interest in this field: for instance, contributing the simulation modules in [6] implementing the Digital Video Broadcasting - Return Channel via Satellite, Second Generation (DVB-RCS2) standard in the NS-3 simulator [7], in order to test and improve the services that can be delivered in supporting IoT/M2M services in RA satellite channels by exploiting such specifications; furthermore, financing research activities on this topic in the context of the Satellite Network of Excellence (SatNEx) IV [8]; last but not least, opening to future research and prototyping activities, such as the upcoming *M2M Maker-Space for Satellite Communications* tender in the ESA Advanced Research in Telecommunications Systems (ARTES) programme [9]. Thus, the road is increasingly paving towards a larger use of satellites in supporting IoT/M2M communications, and this Ph.D. Thesis contributes in highlighting the weaknesses and the strengths of the existing standards and technologies, while proposing guidelines to improve the achievable performance level and several innovative techniques to more efficiently deliver IoT/M2M services via satellite.

1.1 Contributions of the Thesis

Several works can be found in literature on RA schemes designed to support M2M traffic over satellite links; for instance, the Contention Resolution Diversity Slotted Aloha (CRDSA) [10] and CRDSA++ [11] synchronous protocols, or the Enhanced Spread Spectrum Aloha (E-SSA) [12] asynchronous protocol, a core part of the S-band Mobile Interactive Multimedia (S-MIM) standard [13]. These RA schemes are designed to support a large number of

devices, the latter being a typical requirement of M2M scenarios. The main advantage of RA schemes, with respect to Dedicated Access (DA) schemes, is the immediate availability of the channel for data transmission, avoiding any reservation delays, typical of dedicated access schemes. The immediate availability of the channel for data transmission comes at the price of *collisions*, which erase data and trigger a retransmission (if needed). In literature, lots of works have addressed the challenge of performance improvement in RA satellite channels, but the same cannot be said about the investigations on the interactions among different layers of an IoT/M2M protocol stack via RA satellite channels.

The interactions among different layers of the protocol stack in use can have undesired effects on the performance achieved in presence of RA schemes: for instance, if an Automatic Repeat reQuest (ARQ) algorithm is active at MAC layer and a reliable protocol is in use at upper layers, two feedback loops would be active at the same time. The cross-layer interactions can reduce the achievable throughput by triggering unnecessary duplicate retransmissions: this would be the case of a MAC ARQ algorithm coupled with Transport Control Protocol (TCP)-based application protocols or, alternatively, when User Datagram Protocol (UDP)-based application protocols implement reliability features at the application layer. The former case is equivalent to a specific scenario under consideration in this Thesis, i.e., the use of the MQTT (TCP-based) protocol in a DVB-RCS2 environment, while the latter is equivalent to the use of the Constrained Application Protocol (CoAP) (UDP-based) protocol in the same satellite scenario; these two protocols are the most used solutions in terrestrial IoT/M2M deployment, nowadays. Both M2M/IoT protocols are described in the following of this Thesis, and the cross-layer interactions between MQTT and an underlining RA scheme in a DVB-RCS2 exemplary scenario are analytically modelled and empirically evaluated. The use of CoAP is not directly taken into account here, because the reliability feature is only optional in the protocol specifications at this time, and, if enabled, it would rely on a simple (and low performing) Stop-and-Wait ARQ algorithm [14], which would be a performance bottleneck.

This Thesis focuses on modelling the achievable performance of IoT/M2M protocol stacks via RA satellite channels: this analysis binds together the

specificity of the satellite and the desiderata of *interoperability*, which represents an essential feature of present and future IoT/M2M scenarios. More specifically, the contributions provided by this Ph.D. Thesis can be divided into two main research tracks: (*i*) the maximization of throughput in RA satellite return channels and (*ii*) the analysis of the cross-layer interactions between TCP-based application protocols and the CRDSA/CRDSA++ protocol in DVB-RCS2 environments. Massive IoT/M2M scenarios require load control algorithms, centralized or distributed, to allow a large number of RCSTs to contemporarily access the satellite channel. In typical conditions, a RA satellite channel operates in low/medium load conditions, in order to limit the Burst Loss Rate (BLR) due to collisions. A centralized load control algorithm is proposed in this Thesis, exploiting the use of linear codes prior to transmission, in order to achieve a throughput value close to the maximum achievable one in a wide range of load conditions, counteracting the erasures with dynamic adjustments of the average coding rate in use by RCSTs. Further to this, an innovative hybrid access scheme is proposed, in order to allow the coexistence of M2M and non-M2M RCSTs in the same satellite network; the hybrid protocol complements DA to RA schemes, in order to exploit, at the same time, the advantages that both strategies may offer. The second track is focused on the use of the MQTT protocol, an implementation of the Publish / Subscribe (PUB/SUB) paradigm, to deliver IoT/M2M data via RA satellite links.

DVB-RCS2 standard dictates a normative load control algorithm to ensure the stability. Anyway, the latter load control algorithm may need a separate tuning per applicable scenario, which can prove not trivial at all [15]; further to this, the algorithm should be redesigned to actually support realistic Quality of Service (QoS) requirements[2]. Because of this, this Thesis explores a different road: the use of the MQTT protocol (or equivalent TCP-based solutions) to efficiently deliver IoT/M2M data. Thus, the cross-layer interactions between TCP and CRDSA are analytically modelled and empirically evaluated, leading to the conclusions that an TCP-based scenario may perform (at

[2]A comparison with closed commercial protocols, such as Inmarsat BGAN (Broadband Global Area Network) solution for M2M services via satellite, is out of the scope of this Thesis; furthermore, the unavailability of detailed technical specifications may prove impossible any simulated performance comparison.

least) similarly than a UDP-based one used in conjunction with DVB-RCS2 load control algorithm, while proving much easier to deploy. Furthermore, such configuration offers several advantages: for instance, the presence of the *broker* entity ensures a system behaviour almost identical to the presence of a Split TCP-PEP (Performance Enhancement Proxy), typically in place for TCP services via satellite. The *decoupling* between data producers and data consumers, thanks to the broker node, makes also easier the integration among terrestrial and satellite network segments, leading towards the objective of a seamless integration.

1.2 Organization of the Thesis

The structure of this Ph.D. Thesis is the following: Chapter 2 introduces the IoT/M2M fundamentals, describing the most recent services, applications and protocols in this field; furthermore, the challenge posed by this kind of traffic to the existing networks is highlighted. Chapter 3 provides the state of the art of Random, Hybrid and Dedicated access schemes for satellite links, which represent a fundamental part of the overall scenario under consideration in this work; this part provides an overview about the most recent synchronous techniques, with a focus on the DVB-RCS2 standard, whose specifications are taken as reference in the following of the Thesis. Then, Chapter 4 contains the first contributions provided by this work: a load control technique that maximizes the throughput on a satellite RA link in presence of a variable traffic profile, exploiting the use of linear codes prior to transmission; further to this, an innovative hybrid access scheme is designed, aiming at complementing DA schemes to RA ones in presence of traffic peaks, to allow the coexistence among M2M and non-M2M RCSTs. Chapter 5 studies the cross-layer interactions between TCP-based IoT/M2M application protocols and an underlining satellite RA link, focusing on throughput and completion time metrics in different scenarios; furthermore, Chapter 5 provides an analytical model to estimate the TCP throughput, showing a lower error than other analytical models in literature; finally, a simple but effective model that binds together layer-2 and layer-4 loss events is proposed and validated. Finally, Chapter 6 describes a potential application of Network Coding (NC) techni-

ques in satellite networks, when M2M traffic is broadcasted towards Vehicular Ad-Hoc Networks (VANETs) in urban areas: here, the main objective is in ensuring the largest geographical coverage without installing any fixed equipments on the ground, while overcoming the issues that the skylines of cities may pose to a correct reception of the satellite signal, which can be overcome thanks to the use of vehicular cooperation. Eventually, the Conclusions resume the main contributions of this Ph.D. Thesis, as well as the planned future works.

1.3 Publications

In the course of this Ph.D. thesis, the results of my research have been published as follows:

1. Bacco M., Cassarà P., Ferro E., Gotta A., *Generalized Encoding CRDSA: Maximizing Throughput in Enhanced Random Access Schemes for Satellite*, EAI Endorsed Transactions on Mobile Communications and Applications, volume 2, December 2014 [16]
2. Bacco M., Gotta A., Luglio M., Roseti C., Zampognaro F., *A new RA-DA hybrid MAC approach for DVB-RCS2*, in IEEE BlackSeaCom International Black Sea Conference on Communications and Networking (Constanta, Romania, 18-21 May 2015). Proceedings, pp. 215 - 219 [17]
3. Bacco M., De Cola T., Giambene G., Gotta A., *Advances on Elastic Traffic via M2M Satellite User Terminals*, IEEE 12th International Symposium on Wireless Communication Systems (ISWCS), 25–28 August 2015, Brussels [18]
4. Giambene G., Muhammad M., Luong K., Bacco M., Gotta A., Celandroni N., Susanto J., Hu F., Pillaii M. A., de Cola T., *Network Coding Applications to High Bit-Rate Satellite Networks*, 7th EAI International Conference on Wireless and Satellite Systems, Bradford, 6-7 July 2015 [19]
5. Bacco M., De Cola T., Giambene G., Gotta A., *M2M traffic via Random Access Satellite links: interactions between Transport and MAC layer*, IEEE/ACM Transaction on Networking (submitted)

6. Bacco M., Gotta A., *RLNC in Satellite Networks: A Cooperative Scenario for Delivering M2M Traffic*, International Journal of Satellite Communications and Networking (submitted)

In addition to that, I participated in several research projects and activities that lead to the following publications:

1. Bacco M., Cassarà P., Ferro E., Gotta A., *Generalized Encoding CRDSA: Maximizing Throughput in Enhanced Random Access Schemes for Satellite*, Personal Satellite Services, pp. 115-122, Springer International Publishing, Toulouse, France, June 27-28, 2013 [20]

2. Bacco M., Gotta A., Roseti C., Zampognaro F., *A study on TCP Error Recovery Interaction with Random Access Satellite Schemes*, IEEE 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), pp. 405-410, 8-10 Sept. 2014, Livorno [21]

3. Bacco M., Ferro E., Gotta A., *UAVs in WSNs for Agricultural Applications: an Analysis of the Two-Ray Radio Propagation Model*, IEEE Sensors conference 2014, pp. 130-133 [22]

4. Bacco M., Ferro E., Gotta A., *Radio Propagation Models for UAVs: what is missing?*, ACM 11th International Conference on Mobile and Ubiquitous Systems (MOBIQUITOUS): Computing, Networking and Services, pp. 391-392, London, 2–5 Dec. 2014 [23]

5. Bacco M., De Cola T., Gotta A., *TCP New Reno over DVB-RCS2 Random Access Links: Performance Analysis and Throughput Estimation*, IEEE GLOBECOM conference, 6-10 December 2015 [24]

6. Ali M., Bacco M., Celandroni N., Giambene G., Gotta A., Fun Hu Y., Kinyuy Jaff E., Anh Le V., Kim Luong D., Muhammad M., Pillai P., Susanto M., *SatNEx IV - Tecnical Note of SatNEx IV - Call of Order 1 Part 1 – Work Item 4: Network Coding Applications in Satellite Communication Networks*

Chapter 2

Internet of Things and Machine To Machine: Services, Applications and Protocols

M2M communications describe a set of technologies to provide wireless/wired devices of the same ability to exchange information without an explicit human intervention. Here, the term *devices* can refer to hardware or to a piece of software. M2M communications were born in 1980s, when Industrial Control Systems (ICSs) were born: in fact, one of the most common use was inside the industrial plants, where wired systems were used to monitor and report about the status of running machines used in the industrial production process. They represent the early form of M2M communications. The collected information was available to human operators to allow an efficient supervision, but some systems were also able to *actuate* programmed actions in order to control the monitored process. In this context, devices can be seen as *sensors* and *actuators*, i.e., the former being able to sense physical dimensions and the latter being able to operate on the environment, in order to modify its status according to the data collected by the M2M sensors. A characteristic, among others, is here worth of mentioning: the absence of *interoperability*,

because each manufacturer sold his Supervisory Control and Data Acquisition (SCADA) system, which unlikely operated in conjunction with another manufacturers' devices.

Years later, Internet was born, along with its services, applications and protocols to come. Internet allows connecting together PCs at that time, together with servers hosting a specific service. During these years, the large number of possibilities offered by such an enormous connected network was unveiled: still nowadays, new domains open to the users and to the market thanks to Internet. Even resource constrained devices are now able to connect to Internet and to exchange data with other similar devices. Several different *things* are able to share and exchange information: the so-called IoT was mentioned for the first time between 2008 and 2009 [25]. Since then, the attention has been focused on IoT and its incredible variety of application fields. Figure 2.1 shows an partial overview of existing and future application fields involving the IoT paradigm, visualizing how almost all the *things* in our lives can become interconnected, thanks to Internet. M2M communications reach the final users thanks to IoT: in fact, on a very general note, IoT shares the same idea behind M2M communications, plus Internet connectivity. A key characteristic of IoT is surely *interoperability*: the proprietary solutions offered by the different manufacturers are now replaced by standard solutions, relying on open protocol stacks. A large industrial and scientific effort is still in place, in order to allow *things* to operate without any human intervention. The use of standardized protocol stacks makes easier to integrate different products sold by different manufacturers. IoT/M2M communications are a hot topic for the future of computing and communication networks: not only they represent a technological revolution that already have a profound impact on our daily lives, but they also create a new *ecosystem*. In the following Sections, a brief overview of IoT/M2M services, applications and protocols is proposed.

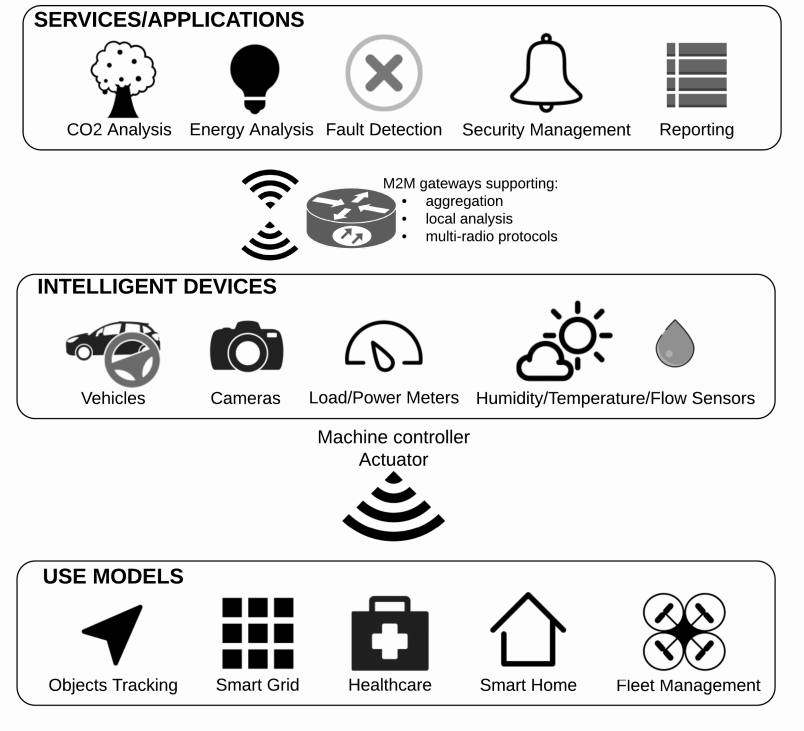


Figure 2.1: An overview of the application and services in IoT/M2M ecosystems

2.1 A brief overview of IoT/M2M-based services and applications

As already shown in Figure 2.1, the potentialities offered by IoT make possible a huge number of services and applications: part of them are already available to the final users, others will be so in the near future. The impact on our lives is visible and grows day by day: the things surrounding us are now becoming *connected things*: power plugs, lamps, phones, readers, fridges, washing machines, and so on. But also outside our home the interconnection is growing: for instance, while travelling, when sick, at work. The main application areas can be grouped in the following domains [26]:

- transportation and logistics: assisted driving, augmented maps, and so

on;

- healthcare: tracking and sensing, and so on;
- smart environment: smart museums, gyms, houses, and so on;
- personal and social: thefts countermeasures, social networking, and so on.

Some distinctive characteristics of IoT/M2M communications have been identified [27, 28]:

- multitude: the more impacting change brought by IoT/M2M. The number of connected IoT/ M2M devices may have already exceed the sum of the devices which directly interact with humans. The magnitude of this change is something probably never experienced before, putting a large pressure on network architectures and on the load offered to networks, designed to accommodate fewer *actors* and different types of traffic. The communication protocols in use have been designed to satisfy different requirements, thus innovative solutions are to be found;
- scalability: because of the multitude of devices, it is imperative that the protocols in use can be easily and gracefully adjusted to varying node densities, exchanging little or no control data, and maintaining fairness;
- variety: considering that each field is gradually replacing its *things* with *connected things*, the possible use cases of IoT/ M2M are exponentially growing. It means that a large variety of devices is emerging, in terms of data exchange rate, form factor, computing and/or communication capabilities. The most critical factor is the *heterogeneity*, which can represent a major challenge to the *interoperability*;
- intrusiveness: the number of installed IoT/M2M devices is growing, and they are being installed everywhere. In turn, it leads to privacy issues: the advantage offered by a better management of monitored systems is in contrast with the necessity to protect end-users data, collected by these devices, able to offer a deep and rich view of users' habits;
- invisibility: no human control should be necessary (the so-called *zero-touch manageability*); it is a strong requirement, protecting both devices

and humans from the overwhelming work it may be required otherwise. It translates into a necessary seamless integration;

- criticality: in the field of e-Health, for instance, devices can be life-savers: reliability is critical, as well as latency, and the latter can seriously challenge the capabilities of current networks;
- low power: these devices can have limited access or none to a power source, thus they must be designed to reduce as much as possible the energy consumption;
- small burst transmission: each device produces, regularly or not, data content which must be collected. While the amount of data generated from a single device can be really low, the same cannot be said for the aggregate traffic. This aspect is deepened in Section 2.2 because of its importance;
- coexistence: these devices are operating in the unlicensed bands and will likely do so in the future. Considering the widespread deployment, a coexistence issue needs a carefully design to reduce the impact on existing traditional networks operating in the same bands (e.g., WiFi and Bluetooth);
- cost: finally, in order to make IoT/M2M scenarios a true reality for final users, the devices must be cost effective. Even if complex devices can have desirable properties, the final price may not be affordable; it means that a large number of small and cheap devices is expect to be in place, each of them being able to accomplish small and easy tasks.

Because of its vast diffusion in every field of our life, IoT/M2M communications are generating huge amounts of traffic, which has characteristic features worth describing, as in the Section 2.2.

2.2 Traffic features of IoT/M2M applications

The distinctive traffic features of IoT/M2M applications are clearly described in [3]: the patterns may include periodic, event-driven and multimedia streaming profiles, depending on the specific application. Even if developing a

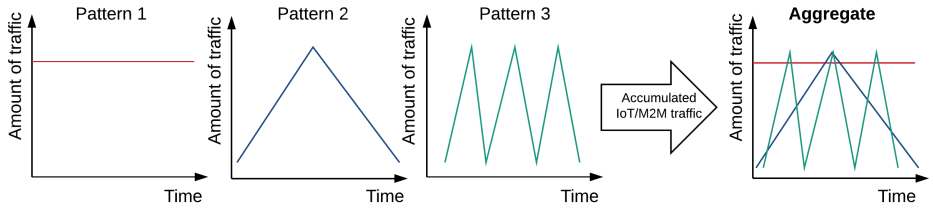


Figure 2.2: An example of a complex aggregate IoT/M2M traffic profile obtained as superposition of several simple traffic patterns exhibited by different services/applications.

simple model able to describe the traffic profile generated by a specific application can be an easy task, the same cannot be said while looking at the aggregate pattern, as generated by a large population of IoT/M2M devices. The issue posed by potentially really large volumes of data must be carefully addressed to guarantee needed QoS and the reliability of applications. Figure 2.2 shows how the superposition of traffic profiles coming from different applications accumulates in the network, generating a complex traffic pattern. The aggregate traffic pattern is also function of the protocol stack in use: some protocols rely on congestion control algorithms, thus they react in a dynamic way to the experienced network conditions, while others can show simple and stationary behaviours. Because of that, different protocols enabling IoT/M2M are discussed in a thorough way in Section 2.3.

2.3 IoT/M2M protocols

The effort for identifying a common IoT/M2M protocol stack is one of the biggest challenge of the last years. The Internet Engineering Task Force (IETF), along with its working groups, and the European Telecommunication Standard Institute (ETSI), are both struggling with the design of a protocol stack solution able to satisfy the aforementioned requirements, while keeping the protocols as simple and lightweight as possible, in order to allow the implementation also on resource constrained devices. Market companies are proposing alternative solutions, too, targeting specific customer needs, thus adding different visions and models in this context. The main objective should

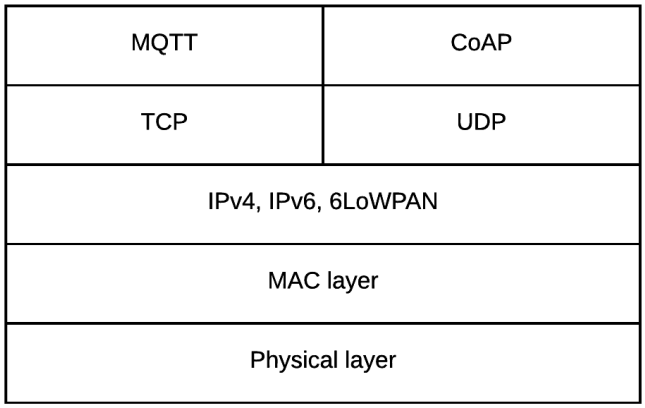


Figure 2.3: Protocol stack of the most common IoT/M2M solutions.

be the integration of resource constrained devices in a interoperable way, ensuring Internet connectivity.

In the following, an overview of the protocols designed for IoT/M2M communications is provided, grouped by communication layer.

2.3.1 Application layer protocols

The application layer is the highest layer in the ISO/OSI model[1]. Currently, two protocols are playing their battle in this field[2]: the CoAP protocol, as proposed by the IETF Constrained RESTful Environments (CoRE) working group and described in Section 2.3.1.1, and the MQTT protocol, developed by IBM in 1999 and described in Section 2.3.1.2. Figure 2.3 shows the different protocol stacks of the two protocols under comparison, while Table 2.1 provides numerical values to quantify the main differences between them.

[1]The Open Systems Interconnection (OSI) model, a product of the Open Systems Interconnection project at the International Organization for Standardization (ISO), is a conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system.

[2]Several different protocols are actually available for use, but the attention is here focused on the ones receiving the largest attention from the market and the standardization organisms.

2.3.1.1 Constrained Application Protocol

The CoAP protocol has been designed by the CoRE working group. The CoRE working group provides a framework for resource-oriented applications intended to run on constrained-node[3] Internet Protocol (IP) networks. CoAP is rapidly gaining popularity: for instance, Open Mobile Access (OMA) Lightweight M2M (LWM2M) is a profile for IETF CoAP and related standards; it specifies a set of common interfaces and data models to enable plug and play interoperability between CoAP devices and local/remote services [29].

The general architecture consists of resource-constrained nodes, called *Devices*, which are responsible of one or more *Resources* in a constrained-node network environment: sensors, actuators, combinations of values, and/or other information. Constrained-node networks can be found as part of home and building automation, energy management, and IoT. CoAP is designed for use between devices on the same constrained-node network, between devices and general nodes on the Internet, and between devices on different constrained-node networks. CoAP, which implements the Request/Response communication paradigm (optionally it supports the so-called *observer pattern*, i.e., a *push* strategy for delivering messages, opposed to the default *pull* strategy), conforms to the Representational State Transfer (REST) style: it abstracts all the objects in the network as resources. Each resource corresponds to an unique Universal Device Identifier (URI) from which the resources can be operated stateless. CoAP realizes a subset of HTTP protocol, adding other features, as asynchronous message exchange. Unlike HTTP, CoAP is UDP-based[4]: to realize reliable communications, a two-layer structure is introduced, offering asynchronous operations with UDP and confirmable (CON), non-confirmable (NON), Acknowledgement (ACK) and reset (RST) messages. Reliability and congestion control are then performed at the application layer: the former feature is optional, and currently relies on the implementation of a simple and low-performing Stop-and-Wait ARQ

[3]Request For Comment (RFC) 7228 defines the *Terminology for Constrained-Node Networks*. The constraints may include: low achievable bit-rate/throughput; high packet loss; highly asymmetric links; presence of sleeping devices; lack of common services (ex. multicast).

[4]A draft IETF proposal exists to implement CoAP over TCP: `https://datatracker.ietf.org/doc/draft-ietf-core-coap-tcp-tls/`

algorithm, allowing to use more advanced mechanisms if also dealing with possible congestion phenomena.

CoAP can be shortly described by its main features:

- constrained WEB protocol fulfilling M2M requirements;
- asynchronous message exchanges;
- low header overhead and parsing complexity;
- a stateless HTTP-CoAP mapping, allowing proxies to provide access to CoAP resources via HTTP in an uniform way (and viceversa).

CoAP typically works in conjunction with the IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) [30] and IP protocol suite, as shown in Figure 2.3.

2.3.1.2 MQTT

MQTT is an application-layer protocol developed by IBM in 1999. It conforms to the PUB/SUB paradigm. Three logical entities can be described in this communication paradigm: the *publishers*, acting as data sources and periodically sending data to the *broker*, a central entity that collects them, and the *subscribers*, which declare their interests in one or more *topics* and receive data from the broker according to the said interests. Figure 2.4 depicts the PUB/SUB communication paradigm. This paradigm decouples *producers* and *consumers*, which do not know each others. In this way, the complexity is moved inside the broker (or *rendezvous*, in the Information-Centric Networking (ICN) terminology), the only logical entity which keeps track of the topics that are *declared* by the publishers. The broker is fed by the publishers; furthermore, the broker keeps track of the interests declared by the subscribers, of their IP addresses and, finally, takes care of the data delivery. MQTT is a TCP-based protocol, thus the reliability is guaranteed at transport layer. Additional layers of reliability can be used at the application layer, for the application to be aware of the correct delivery of data. MQTT has a larger application header than CoAP, and, in general, it exhibits a larger overhead due to the use of a reliable transport protocol. Table 2.1 proposes a simple but meaningful comparison between CoAP and MQTT protocols.

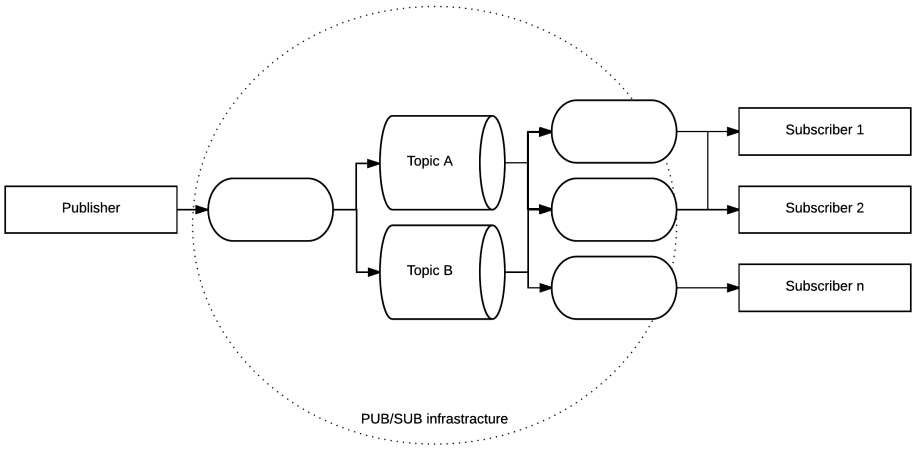


Figure 2.4: A logical description of the PUB/SUB communication paradigm.

Table 2.1: A brief comparison between CoAP and MQTT. The sizes of the transport headers are typical values.

Parameters	**CoAP**	**MQTT**
Application header [B]	4	5
Transport header [B]	8	32
Max. message size [B]	1280	virt. unlimited
Reliability	application layer (opt.)	transport layer

2.3.2 Transport layer protocols

A new conception of transport layer protocols may be required to proper support IoT. The major goals of the transport layer are to guarantee end-to-end reliability and to perform end-to-end congestion control. In the traditional Internet, TCP is in use at transport layer for reliable communications, but it shows several shortcomings when applied IoT scenarios. For instance:

- connection setup: or the so called Three Way Handshake (3WHS). It adds a not negligible overhead to IoT communications, because the amount of user data to be delivered is often equal to the control traffic

exchanged because of 3WHS; the latter procedure also delays the data exchange because of the steps of connection set-up;

- congestion control: the way TCP works makes less likely that the network can collapse because of congestion events; but that mechanism can be useless for small amount of data, because the whole TCP session is likely to be concluded by transmitting one or few data segments, thus the Additive Increase Multiplicative Decrease (AIMD)[5] algorithm can be of little help under these operating conditions;
- buffering: TCP requires the data to be stored in buffers at source and destination for retransmission and reordering, but buffers can be costly in resource constrained devices.

MQTT is TCP-based, and it represents one of the most common solutions in current deployments. It means that, nowadays, the load offered by IoT is sensibly impacting on existing networks. New solutions may be required to alleviate that, which require the design of innovative transport protocols, able to friendly interact with existing flows to guarantee the fairness and the respect of existing and upcoming QoS requirements.

2.3.3 Network layer protocols

The Routing Protocol for Low Power and Lossy Networks (RPL) protocol, developed by the Routing over Lossy and Low-Power Networks (RoLL) working group, has been proposed by IETF as network layer protocol. RPL is focused on the routing protocol design and is committed to the standardization of the IPv6 routing protocol for Lossy and Low-Power Networks (LLNs) [31]. It is a distance-vector routing protocol, and a performance evaluation is available in RFC 6687. Briefly, RPL supports multipath, thus it is able to handle packet reordering, but it performs poorly in terms of energy efficiency [31] because of little attention to the duty cycling during the design phases. The addressing scheme in IPv4 does no provide scalability and, in order to solve that, IPv6 has been introduced. IPv6 over 6LoWPAN provides a set of protocols that can be used to integrate resource-limited devices into IPv6 networks by simplifying IPv6.

[5] AIMD is the feedback control algorithm used by TCP.

2.3.4 MAC layer protocols

The MAC layer protocols for wireless environment have received large attention in literature. A simple and broad classification of these protocols is the following one: *contention-based* protocols, discussed in Section 2.3.4.1, *contention-free* protocols, discussed in Section 2.3.4.2, and *hybrid* protocols, discussed in Section 2.3.4.3.

2.3.4.1 Contention-based protocols

The contention-based protocols, or RA protocols, are the simplest available protocols in terms of set-up and implementation: any terminal willing to transmit data can immediately initiate the transmission, contending for the use of the channel. If Collision Avoidance (CA) techniques are in use, then the data exchange can start only if the channel is not occupied by another transmission, otherwise the transmission is delayed to reduce the probability of *collisions*. If CA techniques are not in use or they cannot be used because of the characteristics of the channel (for instance, in satellite scenarios), a collision can occur. The latter phenomenon occurs when two or more nodes transmit data in the same time interval (if using Time Division Multiple Access (TDMA) protocols) or at the same frequency (if using Frequency Division Multiple Access (FDMA) protocols); the data cannot be recovered at the receiver because of the destructive interference among the transmitters. The first and notable example of a contention-based protocol is ALOHA [32], and its slotted variant, namely Slotted Aloha (SA) [33], which allows a data exchange to start as soon as new data is ready for transmission. ALOHA and SA exhibit high collision rates, mainly at medium and high loads, which severely impact on the achievable throughput. ALOHA and SA throughput can asymptotically reach 18% and 36%, respectively. The most common RA protocol for M2M communications in terrestrial networks is the Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 protocol. It is based on the of Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) scheme. 6LoWPAN represents the enabling adaption layer for IoT/M2M networks over the IEEE 802.15.4 protocol stack, where the use of IPv6 is foreseen. This solution is designed for resource constrained devices and it is opposed to

the so-called *full-IP* solution.

The use of Carrier Sense Multiple Access (CSMA) scheme can reduce the collision rate and it is widely adopted in several protocol implementations: for instance, IEEE 802.11 protocol family is based on the use of CSMA scheme [34]. In satellite networks, the use of Carrier Sense and CA techniques is not possible, because terrestrial terminals are sparsely placed on the ground and are far away from the intended receiver, which is reached via a satellite link: thus, a Satellite Terminal (ST) cannot *sense* the channel prior to transmitting, in order to avoid collisions. A deepening on the satellite RA schemes is provided in Chapter 3, because they represent a fundamental part of this Ph.D. thesis.

2.3.4.2 Contention-free protocols

Contention-free, or DA, protocols eliminate the issue of collisions by preallocating resources to the terminals. In this way, a fraction of the time (or a fraction of the frequency bandwidth) is allocated to a node in a temporary or permanent way, and the other nodes in the network cannot use it. While the bandwidth utilization is maximized if contention-free protocols are used, poor flexibility must be faced, which is a limitation in very dynamic networks. IoT/M2M networks are highly dynamic and densely populated networks; furthermore, short data bursts are expected. Because of these characteristics, DA protocols usually perform poorly in IoT/M2M scenarios: the static allocation of a resource must be handled by a central entity and requires a set-up time, which is not compatible with the needed dynamicity. The drawbacks of DA protocols in IoT/M2M scenarios outperform the advantages they provide, when opposed to the use of RA protocols.

2.3.4.3 Hybrid protocols

Hybrid protocols are designed to jointly exploit the advantages offered by RA and DA schemes: the high throughput guaranteed by the latter ones and the support for dense and dynamic populations offered by the former ones. Hybrid protocols switch their operating conditions according to the offered load, for instance: at low loads, RA protocols are preferred because of their simplicity and absence of set-up times; at high loads, DA protocols

may represent a better choice because they can offer higher throughput and absence of collisions at the cost of a limited flexibility. This class of protocols probably represent the most promising one for IoT/M2M communications, where the reliability in data delivery should be guaranteed as much as possible, but simplicity is of the essence, too, in presence of a multitude of resource-constrained devices.

2.3.5 Considerations and discussion

In this Chapter, an overview of the most popular IoT/M2M protocols has been provided, together with an overview of the plethora of services and applications the market offers in this segment. The main characteristics of this existing and rapidly evolving new type of communications have been highlighted, as well as the issues they pose to the existing networks. The efforts put in the standardization by international working groups and communities, for instance IETF, are described, together with the most recent proposed standards, in order to guarantee coexistence and interoperability among different devices and different manufactures implementations. Finally, the focus has been put on the MAC protocols currently used, in order to highlight the requirements and the available solutions at the MAC layer, where RA and hybrid protocols are the most promising solutions to the challenges that the IoT/M2M paradigm poses. A deepening on the most used satellite protocols is in Chapter 3, in order to focus the attention on the specific characteristics of satellite channels, which may require different approaches w.r.t. the ones in use in terrestrial scenarios.

Chapter 3

Background: Random, Hybrid and Dedicated Access Satellite Schemes

In this Chapter, the most popular RA, DA and hybrid schemes for satellite communications are described, as a necessary background to support the contributions in the next Chapters. Note that geosynchronous (GEO) satellites communications are here taken into account, even if the protocols later described may be also used in different operating conditions. In satellite networks, the use of Carrier Sense and CA techniques is not possible, because terrestrial terminals are (typically) sparsely placed on the ground and are far away from the intended receiver(s), which is reached via a satellite link: thus, a ST cannot *sense* the channel before transmitting, in order to avoid collisions. Because of this, the satellite protocols are mainly based on Aloha protocol and its evolutions, or they use static resources assignment made by a centralized entity.

3.1 Random Access Schemes

This Section provides an overview on the evolution of the RA protocols based on Aloha, highlighting the key characteristics of each solution and the achie-

vable performance. This Chapter only surveys synchronous RA protocols, i.e., when a common reference clock is used to synchronize the ground stations, with the notable exception of the Aloha protocol, which is an asynchronous one. The asynchronous protocols, as, for instance, E-SSA [13] or Asynchronous-CRDSA [35], are not part of this Ph.D. thesis because the focus is on RA schemes applicable in DVB-RCS2-based scenarios.

3.1.1 Aloha

Aloha protocol, a pure asynchronous one, has been introduced in 1970 by Abramson [32]. It represents one of most simple schemes for RA communications, where any ground station, or ST, transmits its data as soon as they are available. The simplicity of such a protocol is counterbalanced by the *collision* phenomenon: when two or more STs transmit data in an overlapped manner, the bursts collide and one or more retransmissions are needed to recover the lost data. If ARQ techniques are used, each ST waits for an ACK after each data transmission: in case it is not observed, then the data burst is assumed to be lost. The station waits a time interval known as *back-off time* before retransmitting, in order to reduce the probability of a new collision. A collision can be partial or complete: if the transmissions are completely overlapped, the collision is said to be *complete*, while it is said to be *partial* if only part of the transmissions is overlapped. In both cases, the collided bursts are erased, i.e., dropped at the receiver, because the decoding phase fails.

$G = N/n$ is the normalized load in presence of N STs transmitting in a frame composed of n time-slots. G^* is the normalized load level offering the peak throughput. The maximum throughput Aloha protocol can offer is $T \approx 0.18$ at $G^* \approx 0.5$ [32].

3.1.2 Slotted Aloha

In 1972, Roberts introduces the SA protocol [33]. The main difference with respect to Aloha protocol is the fact that the time is *slotted*, i.e., the ground stations are synchronized in time, by relying on a broadcast reference clock, and they can only transmit data in *time-slots*, pre-defined time intervals. In this way, the collisions can erase the transmissions only completely: partial

collision are avoided by the synchronization expedient. The use of the synchronisation allows for a larger normalized throughput $T \approx 0.36$ at $G^* \approx 1$, as visible in Figure 3.1, where the throughput achievable by SA is compared with the throughput achievable by Diversity Slotted Aloha (DSA), described in Section 3.1.3.

3.1.3 Diversity Slotted Aloha

DSA is an enhancement of the SA protocol, proposed in 1981 by Choudhury and Rappaport [36]. The core idea is the use of *replicas*, i.e., two or more copies of the original burst are sent on the satellite channel. The replicas are sent either on different frequency channels or spaced apart by random time intervals. The use of replicas increases the probability of a correct reception of the original burst because it is sufficient to correctly decode one of the replicas to retrieve the original information, thus reducing the number of required retransmissions to eventually deliver a burst. The use of two replicas offers the largest achievable throughput offered by this RA schema, which is also larger than the one offered by SA at low loads, as visible in Figure 3.1. Then, the throughput decreases rapidly. The use of a larger number of replicas offers more robustness at the cost of a reduced throughput. Thus, DSA offers better performances than SA at low traffic loads, reducing the delay needed to correctly deliver a burst, but the throughput is lower at medium/high traffic loads. Furthermore, it must be noted that, for the same peak transmission power of the SA scheme, the average transmitted power of DSA is doubled [37].

3.1.4 Contention Resolution Diversity Slotted Aloha (CRDSA) and CRDSA++

CRDSA has been introduced in 2007 by Casini et. al. [10]. This innovative protocol is a further enhancement of DSA, implementing the Successive Interference Cancellation (SIC) algorithm to achieve a larger throughput than SA and DSA. It relies on the use of replicas, whose number is fixed and equal for each ST, as in DSA, but each replica has pointers to the position of the other copies in the frame. When a replica is correctly decoded at destination,

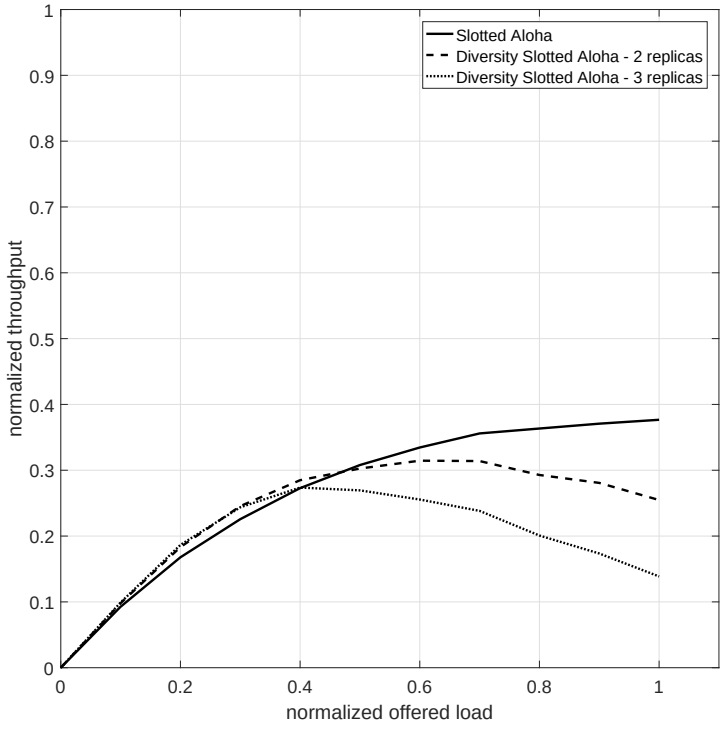


Figure 3.1: Normalized throughput of SA and DSA - 64 time-slots per frame.

the other ones can be *subtracted*, i.e., *interference cancellation* [10] can be applied, in order to subtract the decoded signal from the time-slots containing the replicas, thus increasing the probability to recover any other bursts transmitted in the same time-slot(s). SIC algorithm runs iteratively, until no new bursts can be decoded, then it stops and the decoding procedure of CRDSA frame terminates.

Thanks to the use of SIC, CRDSA offers a large performance gain with respect to SA and DSA. Figure 3.2 shows the throughput that CRDSA can achieve with different number of replicas: it shows a quasi-linear trend up to approximately twice the normalized offered load of DSA. In fact, $G^* \approx 0.7$ if using a frame composed of 64 time-slots. When using 3 replicas, T is

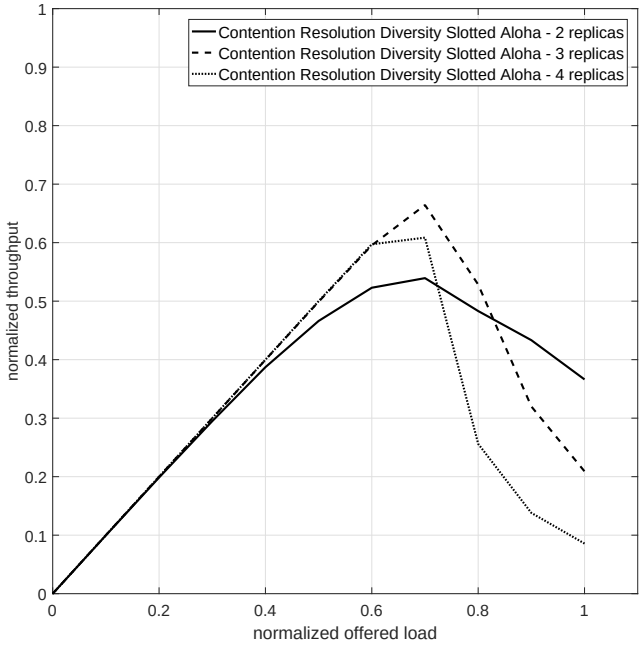


Figure 3.2: Normalized throughput of CRDSA - 64 time-slots per frame.

higher than with any different number of replicas (for a maximum frame length as allowed by DVB-RCS2, i.e., [64, 128] time-slots), and that is why CRDSA is usually configured with 3 replicas, in a configuration referred to as CRDSA++ [12] (more precisely, CRDSA++ allows sending more than two replicas). Figure 3.3 shows a CRDSA++ frame and how the SIC algorithm works: a replica of the user 1's data burst can be decoded in time-slot 4, and then the copies in time-slots 2 e 9 can be cancelled. At this point, user 4's replica in time-slot 9 can be decoded, thus the replicas in time-slots 2 and 7 can be cancelled. Proceeding iteratively on the frame, all the transmissions can be correctly decoded in the example case shown in Figure 3.3.

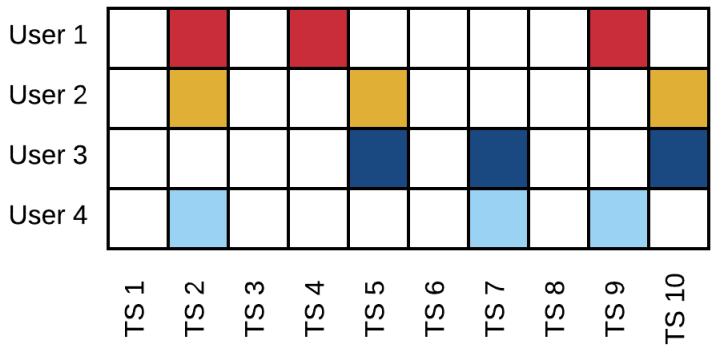


Figure 3.3: Example of a CRDSA++ frame with 3 replicas (TS stands for time-slot)

3.1.5 Irregular Repetition Slotted Aloha

Irregular Repetition Slotted Aloha (IRSA) has been introduced by Liva in 2011 [38]. It relaxes the constraint about the fixed number of replicas, as in DSA and CRDSA: each ST can randomly choose how many copies of the same burst are to be sent. It makes use of the SIC procedure, and the irregular number of replicas offers a small throughput gain over CRDSA. The throughput gain increases as the number of available time-slots increases. IRSA offers an analytical asymptotic throughput $T = 0.92$ at $G^* = 0.92$; in usable configurations, IRSA offers a throughput $T = 0.78$ at $G^* = 0.8$, when a frame composed of 200 time-slots is used.

3.1.6 Coded Slotted Aloha

The Coded Slotted Aloha (CSA) protocol has been introduced by Paolini et. al. in 2011 [37]. This scheme allows for a little throughput gain over IRSA, but introduces a linear coding block prior to transmission and a linear decoding block at the receiver, thus increasing the required computational power both at source and destination sides. The main idea behind CSA is that bursts to be transmitted are firstly split into sub-segments and encoded through a local packet-oriented linear code. The data is then transmitted on the RA satellite channel and, on the receiver side, SIC combined with decoding of the local

code is performed to recover from collisions. Each data segment of duration T_{SA} is split into k information sub-bursts of duration $T_{CSA} = T_{SA}/k$; then, the k information sub-bursts are encoded via a packet-oriented linear code which generates n_h encoded segments of duration T_{CSA} each. Each user can randomly choose the linear code (n_h, k) to be used prior to transmission. CSA offers a higher power efficiency when the linear code rate $R = k/n_h$ is high, i.e. $R > 1/2$, while IRSA offers a higher power efficiency if low rate codes are considered, i.e., $R \leq 1/2$. The throughput that IRSA can achieve is $T \approx 0.8$ at $G^* = 0.8$, with an asymptotic value of $T = G^* = 0.9$.

3.2 Dedicated Access Schemes

DA protocols exploit the concept of *reservation*: instead of competing for accessing to a sub-set of resources, the latter can be reserved, temporarily or permanently, for a specific ST to use. These protocols can achieve higher throughput than RA protocols, but a reservation delay must be paid: when a ST has bursts pending for transmission, it issues a reservation request to a coordinating unit, the so-called Network Control Center (NCC), making use of a reservation channel (typically SA-based), which reserves one or more time-slots in the upcoming frames, if any resources are available, and sends back the reservation response. Considering the large Round Trip Time (RTT) that the transmissions experience on satellite links, the reservation process introduces some not negligible delay, which is $\geq$ 520 [ms] in case of GEO satellites. Therefore, the user applications which require immediate access to the channel cannot take advantage of DA techniques, and RA protocols are to be preferred. The most used DA schemes are described in what follows:

- Constant Rate Allocation (CRA): static capacity, allocated as long as the ground station is active, by NCC;
- Rate Based Dynamic Capacity (RBDC): the ground station issues reservation requests by making use of predictive techniques, based on a estimation of the arrival rate of data bursts in the transmission buffer. This scheme is largely used in presence of applications showing constant transmission rates;

- Volume Based Dynamic Capacity (VBDC): ST issues reservation requests derived from the transmission buffer sizes. NCC allocates the requested capacity, if available, and new requests are cumulated to the previous ones. To avoid de-synchronization issues, Absolute VBDC (AVBDC) protocol can be used to periodically resynchronize the allocations;
- Free Capacity Allocation (FCA): if any residual capacity is available in the channel after satisfying all the reservation requests, NCC can allocate it to the ground stations without explicit requests; if combined with some sort of heuristics and traffic models, it can reduce the number of further time-consuming allocation requests.

3.3 Hybrid Access Schemes

The advantages that DA and RA protocols offer can be combined together by designing and making use of hybrid access protocols. The main idea is in differentiating the traffic rate each ST exhibits: if sustained rates are observed, then DA access schemes should be used; instead, if STs show bursty and sporadic traffic rates, then RA schemes exhibit greater potentials, because of the immediate beginning of transmissions while opening to the accommodation of a larger population.

3.3.1 Reservation Aloha

Reservation Aloha [39] proposes a scheme similar to SA, but with the possibility for RCSTs to temporary *reserve* a time-slot previously used in a successfully way. In [39], the frame is supposed to have a duration larger than the maximum propagation delay of the channel, thus RCSTs are aware of the usage status of the time-slots in the previous frame. If RCST A has successfully transmitted in the i-th time-slot in the previous frame, the i-th time-slot is reserved to A as long as it has data to transmit; on the other hand, unused or collided time-slots are open for contention in the next frame. The main drawback of the Reservation-Aloha scheme is the frame duration, which should be larger than 520 [ms], when applied to scenarios involving GEO satellites.

3.3.2 Optimal Adaptive Scheme

The Optimal Adaptive Scheme [40] relies on the so-called *urn protocol* for hybrid access: RCSTs do not have the same *transmission probability*[1] or *access rights*. The protocol defined in [40] assumes the perfect knowledge of the number of nodes with bursts pending for transmission: this turns out to be quite complex to achieve in presence of a large multitude of users showing a bursty traffic profile, where control traffic can become a large fraction of the data flowing in the network. Furthermore, the protocol shows good a performance level in presence of light and heavy traffic, but lags in the mid-range of traffic load; and, usually, a RA channel tends to work at low and medium traffic loads, i.e., in the quasi-linear part of the load vs throughput curve (see Figures 3.1 and 3.2).

3.3.3 Selective Reject Aloha/FCFS Scheme

The Selective Reject Aloha/FCFS (First Come First Served) [41] protocol supports both pure RA and implicit reservation modes of operation and offers a good performance level in the region of medium traffic load. It outperforms SA when comparing the delivery delay, but the packet size is a critical choice: in fact, this protocol splits a message into sub-packets of fixed length. Each sub-packet has its own header to permit the reconstruction at destination side, thus an overhead per sub-packet must be paid. When the length of the message exceeds a given threshold, SA offers a larger channel capacity, i.e., a large number of RCSTs is supported [42].

3.4 DVB-RCS2 standard

This Section briefly introduces the most salient features of the DVB-RCS2 standard [43]. The first version of the standard has been amended in 2011 and currently supports both DA and RA schemes on the return link. The standard dictates mandatory support for MF-TDMA (Multi Frequency - Time Division Multiple Access), thus each transmission is confined into time-slots.

[1] The transmission probability is defined as the probability to transmit in the upcoming frame.

DVB-RCS2 is a technical ETSI standard that defines the specifications for satellite Very Small Aperture Terminal (VSAT) systems, which can provide connectivity without need for any local terrestrial infrastructure. Depending on specific satellite link characteristics and other system design parameters, a DVB-RCS2 link can provide up to 10 [Mbit/s] in uplink and several tens of Mbit/s in downlink to RCSTs. Sections 3.4.1 and 3.4.2 briefly describe RA and DA schemes available in the DVB-RCS2 standard, respectively. It is worth noting that DVB-RCS2 opens to the possibility of implementing hybrid access schemes based on the use of RA and DA schemes dictated by the standard.

3.4.1 Random access schemes

This Section introduces the RA schemes implemented in DVB-RCS2 and the terminology used in the standard. Two RA protocols are available for use: SA and CRDSA (the latter optionally supported), described in Sections 3.1.2 and 3.1.4, respectively. SA is mainly used for control traffic, while CRDSA is the most common choice when user data is sent on a RA return link.

In what follows, the aim is in describing in a simple and clear way how CRDSA is designed and implemented in DVB-RCS2: such a description is needed because this RA scheme is used as reference protocol in Chapter 4 and 5. Briefly, a DVB-RCS2 return link is composed by a sequence of superframes; each superframe is further divided into frames. The whole superframe (or a fraction of it) can be allocated to CRDSA and it is referred to as *RA block*. The latter is a key definition: a superframe sequence contains one or more RA blocks, each in a RA channel. Each RA block is divided into time-slots, or *transmission opportunities*. The number of time-slots can range between 64 and 128 and the maximum duration of a RA block is 150 [ms]; the number of replicas, or *instances*, can range between two and three[2]. The position of the replicas is randomized inside the RA block and SIC is applied at the receiver per RA block. Each RCST can exploit any transmission opportunity (also referred to as *MAC burst* or just *burst*) to send data in a RA block. A RCST

[2]Having a single instance is equivalent to using SA. The use of more than three instances is not possible, according to the current specifications, so the use of CRDSA++ should be considered not compliant to the standard.

can transmit up to s unique payloads per RA block, $s \geq 1$, and each unique payload can occupy one or more time-slots.

3.4.2 Dedicated access schemes

The DA protocols available in DVB-RCS2 are CRA, VBDC, AVBDC and FCA, as described in Section 3.2. The main modifications with respect to the first version of the standard are related to the capacity request generation and signalling. DVB-RCS2 does not define a standard algorithm to generate reservation requests: it is intentionally left out as an implementation decision, but maintaining fairness among different vendors' implementations can be difficult.

Chapter 4

Throughput maximization in RA Satellite schemes

This Chapter contains the first contribution of this Ph.D. thesis: the two works provided in what follows are focused on different ways to increase the *throughput* at MAC layer, here the key metric under consideration.

Section 4.1 describes an innovative RA scheme, namely *Generalized Encoding CRDSA*, which is based on a load control technique ad-hoc designed to maximize the throughput when a satellite network undergoes variable traffic load profiles, common in IoT/M2M scenarios. This algorithm can be used as alternative to the frame length reconfiguration, a technique meant to operate a satellite system close to G^* at low/medium loads. Section 4.2 describes the architecture of an innovative hybrid access scheme, based on the MAC protocols provided by the DVB-RCS2 specifications: please note that the DVB-RCS2 standard opens to the possibility of designing and implementing new access schemes based on the supported protocols. Then, an empirical performance evaluation of this innovative access scheme is provided and discussed.

4.1 GE-CRDSA: Maximizing Throughput in Enhanced Random Access Schemes for Satellite

This Section introduces a possible enhancement of the CRDSA protocol[1], named Generalized Encoding CRDSA (GE-CRDSA). GE-CRDSA protocol aims at improving the throughput T when $G \leq G^*$, playing on the opportunity of transmitting an optimal combination of information and redundancy packets frame by frame. In what follows, the achievable improvement in terms of throughput is shown, when a satellite network undergoes a variable traffic load profile, by performing traffic estimation and an adaptive choice of information and redundancy rates.

4.1.1 Introduction

Satellite access networks provide a wide range of services for civilian and military applications. These services include mobile data transfer, localization, satellite television and Internet web traffic. DVB-RCS2 standard has been introduced as a renewed standard for satellite communications and it offers RA and DA access schemes able to support the aforementioned traffic types. The performance of access schemes are strongly linked to the traffic characteristics and, in particular, to the amount of data to be transferred [44]. The advent of M2M/IoT services is increasing the rate of bursty traffic on Internet: such an evolution of the traffic patterns require adequate measurements on available multiple access protocols for satellite, in order to settle the suitability of either a RA or a DA access scheme. In this Section, a study on a generalization of CRDSA and CSA[2] protocols is presented, showing that the random choice of information and redundancy rates can improve the maximum achievable throughput. In particular, the simulation results show how the design of a load control mechanism can help in obtaining a reasonable level of performances at every load level, reducing the need for a less flexible DA-like approach. GE-CRDSA aims at optimizing the two aforementioned

[1] See Section 3.1.4 for a description of the CRDSA protocol.
[2] See Section 3.1.6 for a description of the CSA protocol.

access schemes, by allowing each RCST to remove the limitation on the transmission of a single data burst per frame, leading to the reduction the queuing delay and to an improvement of the achievable throughput at low/medium loads, typical operating conditions of IoT/M2M scenarios relying on the use of RA schemes.

The rest of the Chapter is organized as follows: Section 4.1.2 describes and evaluates the load estimation algorithm in use, as well as an ideal load control strategy based on Generalized Cross Entropy (GCE) numerical methods [45]. The major contribution of this Chapter is in Section 4.1.2.3, a simple but efficient empirically built load control strategy that works without an a priori heuristic and guarantees a performance gain with respect to the GCE-based load controller and, furthermore, the possibility to effortless adapt to the use of any code set and system configuration. Section 4.1.3 shows the simulation results, where the ideal control strategy and the empirically built one are compared. Finally, the conclusions are drawn in Section 4.1.4.

4.1.2 Problem Definition and System Overview

Let us consider a satellite system with a RA return channel. The frame duration is T_F, and each frame is composed of n time-slots. Let N be a Poisson distributed r.v. that represents the number of active user terminals in the i-th frame. $G = N/n$ is the normalized system load. In any frame, a ST randomly draws one couple (n_c, k_c), to be used to encode data prior to transmission, over the set of the available coding schemes C, according to the Probability Mass Function (PMF) defined by the polynomial:

$$\Omega_{(n_c,k_c)}(x,y) \triangleq \sum_c \omega_c \, x^{k_c} \, y^{n_c}, \tag{4.1}$$

where the weight ω_c is the probability of having k_c information packets and $(n_c - k_c)$ redundancy packets over a n_c-long codeword, for the c-th coding scheme. In each frame, the offered information rate (net traffic rate) is given by $k_c\, G$, while the gross traffic rate is given by $n_c\, G$. A flow control algorithm shall account for monitoring the average traffic load during an observation window and for communicating the weight vectors:

$$\mathbf{\Omega} = \{\omega_c;\ c = 1, ...C\} \tag{4.2}$$

to active STs, in order to maximize the system throughput. The optimization problem which leads to the optimal choice of the load vector in (4.2) has been introduced in [46] and it is here presented in the following Sections.

4.1.2.1 Load Estimation

The load estimation (LE) technique described in [47] is able to estimate the probability that a time-slot is in a given state: idle (I), successful (S) or collided (C), which represent zero, one, or two (or more) bursts transmitted in the same time-slot, respectively. When using SA, given the normalized offered load G and the frame length n, the expected numbers of time-slots per state per frame is defined in [47] as:

$$s_I(G,n) = n(1-p)^{Gn}, \tag{4.3}$$

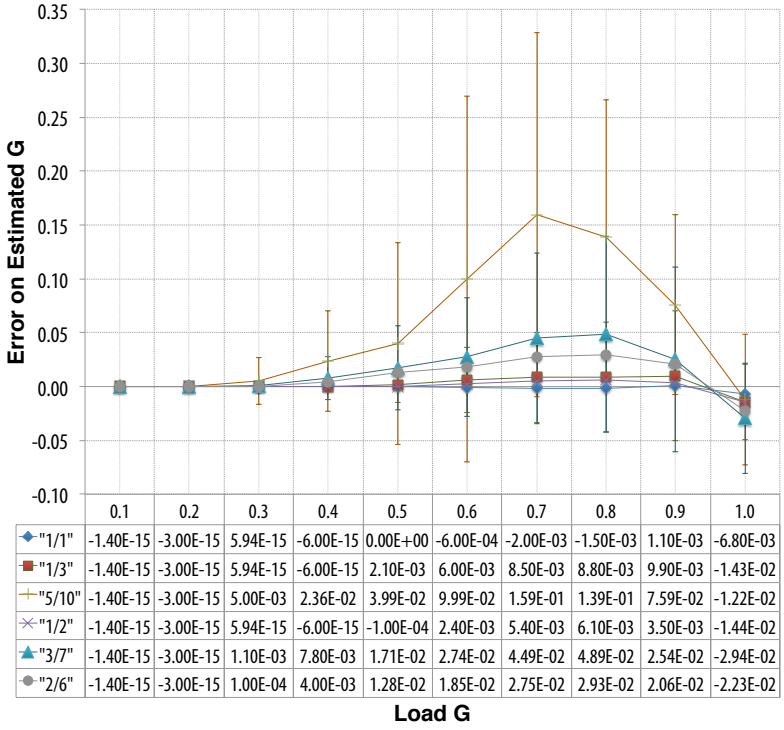
$$s_S(G,n) = Gn(1-p)^{Gn-1}, \tag{4.4}$$

$$s_C(G,n) = n - s_I - s_S, \tag{4.5}$$

where Gn is the number of bursts actually transmitted per frame and $p = 1/n$ is the probability that the burst of a single node is in a given time-slot. This LE technique asses the estimated offered load $\hat{G}$ by inversely tracking equations (4.3)-(4.5) from the observed vector $\mathbf{v} = \{i,\ s,\ c\}$, where i, s, and c are the number of time-slots per frame in I, S, and C states, respectively. Let $\mathcal{H}$ be an LE function that returns $\hat{G}$, which is given in [47] as:

$$\mathcal{H}(\mathbf{v}) = \arg\max_{G} \left(G^s e^{-Gn}(e^G - 1 - G)^c\right) \tag{4.6}$$

where $\hat{G}$ is the estimated load. In order to rely on (4.6) when using CSA and therefore when using GE-CRDSA, $\hat{G}$ must be divided by n_c, because the physical load is *artificially* increased by the linear code length. Therefore, the process of dividing the load estimation by n_c leads to the estimation of the offered load; eventually, to obtain the estimated *normalized* offered load G_{est}, the result must be divided by the number of available time-slots. Figure 4.1 shows the error on load estimation based on simulation results, when using different coding schemes and assuming $n = 100$. Figure 4.1 shows the load estimation error and 0.05, 0.50 and 0.95 quantiles of the values collected



	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
"1/1"	-1.40E-15	-3.00E-15	5.94E-15	-6.00E-15	0.00E+00	-6.00E-04	-2.00E-03	-1.50E-03	1.10E-03	-6.80E-03
"1/3"	-1.40E-15	-3.00E-15	5.94E-15	-6.00E-15	2.10E-03	6.00E-03	8.50E-03	8.80E-03	9.90E-03	-1.43E-02
"5/10"	-1.40E-15	-3.00E-15	5.00E-03	2.36E-02	3.99E-02	9.99E-02	1.59E-01	1.39E-01	7.59E-02	-1.22E-02
"1/2"	-1.40E-15	-3.00E-15	5.94E-15	-6.00E-15	-1.00E-04	2.40E-03	5.40E-03	6.10E-03	3.50E-03	-1.44E-02
"3/7"	-1.40E-15	-3.00E-15	1.10E-03	7.80E-03	1.71E-02	2.74E-02	4.49E-02	4.89E-02	2.54E-02	-2.94E-02
"2/6"	-1.40E-15	-3.00E-15	1.00E-04	4.00E-03	1.28E-02	1.85E-02	2.75E-02	2.93E-02	2.06E-02	-2.23E-02

Figure 4.1: Average error on the load estimation by using [47] when encoding data prior to transmission by using linear codes (n=100). The linear codes shown in the first row of the upper table are in the form (k_c/n_c).

during extensive Montecarlo simulations. The plotted values can also be read in the table in Figure 4.1, where the linear codes are presented as k_c/n_c in the first column. It is visible as the load estimation is subject to an increasing error as k_c increases. In the following, we assume that NCC performs the load estimation and communicates to RCSTs its current value via control messages.

4.1.2.2 GCE-based Load Control Strategy

In order to define a control law able to find the best couple (n_c, k_c) at a given G that maximizes T, a GCE-based controller needs either a priori information, as, for instance, the heuristic shown in Table 4.1 and later discussed, or an

G	**optimal** (n_c, k_c)	T	BLR
0.1	(10, 5)	0.460	$6.598e^{-02}$
0.2	(7, 3)	0.527	$1.222e^{-01}$
0.3	(6, 2)	0.586	$3.288e^{-02}$
0.4	(4, 1)	0.400	$5.00e^{-05}$
0.5	(4, 1)	0.500	$1.23e^{-03}$
0.6	(3, 1)	0.600	$2.93e^{-02}$
0.7	(3, 1)	0.676	$2.952e^{-02}$
0.8	(3, 1)	0.536	$3.083e^{-01}$
0.9	(2, 1)	0.425	$5.319e^{-01}$
1.0	(2, 1)	0.373	$6.361e^{-01}$

Table 4.1: Maximum normalized throughput T when using the best linear code (n_c, k_c) at $G \in [0.1, 1]$, as well as experienced BLR ($n = 100$).

analytical model that provides T as a function of G, k_c and n_c. Reference [10] provides an analytical recursive model able to estimate the probability of correctly decoding a CRDSA transmission (i.e., the throughput per RCST), but only in presence of two replicas (i.e., using a repetition code $(n_c, k_c) = (2, 1)$). It is worth noting that a general analytical formulation to derive BLR is still absent, given as inputs the parameters n_c, k_c, n and G, to the best of the authors' knowledge. Because of this, the following framework is proposed, leading to two alternative options to reach the aforementioned objective: the first solution is based on the use of a priori available data and is discussed in the following, while the second solution is based on an empirically built algorithm and it is shown in Section 4.1.2.3.

Assuming that each RCST can randomly choose the couple (n_c, k_c) to be used prior to transmission in the i-th frame, the throughput maximization

over all the possible couples (n_c, k_c) can be formally stated as:

$$\begin{aligned} &\max\{T\}_{\Omega_{(n_c,k_C)}} \\ &\qquad \text{subject to} \\ &\qquad k_c \leq k_{max} \\ &\qquad n_c \leq n_{max} \\ &\qquad BLR \leq \theta \end{aligned} \tag{4.7}$$

The solution of the optimization problem in (4.7) can be found by solving (4.1), which gives the optimal number of information and redundancy packets to be transmitted in the upcoming frame. The constraints of the optimization problem are the maximum allowed number of information packets k_c and the maximum number of packets n_c per ST per frame and, finally, the maximum tolerable BLR.

An exhaustive evaluation of the throughput achievable by using the best (n_c, k_c) linear codes is available in [46], in order to achieve the maximum T, for $n \leq 1000$; for practical reasons, the main interest is in the range foreseen in the DVB-RCS2 standard, i.e., $n \in [64, 128]$. For instance, Table 4.1 shows the optimal choice of the couple (n_c, k_c) that achieves the maximum T when n=100 and $G \in [0.1, 1]$. In order to use the aforementioned heuristic, it is worth recalling that the load estimation, which leads to the choice of a linear code over another, one may be affected by an estimation error, as shown in Figure 4.1. In addition, the satellite feedback delay makes the estimation obsolete at least of the satellite latency: in case of GEO satellites, $RTT \geq 520$ [ms].

IRSA and CSA performance can be increased in presence of a variable repetition rate. Anyway, being able to control the system and leading its operating point as close as possible to G^* is not a trivial task, because any errors may lead the system to operate in the range $G > G^*$, where the channel stability is not guaranteed [48].

In what follows, it is shown how the problem discussed so far can be addressed by an iterative algorithm based on the numerical methods provided by the GCE methods. In the proposed formulation, C is the finite set of available coding schemes (n_c, k_c) and the c-th coding scheme can be drawn with probability ω_c. The problem can be stated as follows: finding the optimal PMF,

as defined in (4.1), in order to maximize the objective function $\mathbf{O}_T((n_c, k_c))$, respecting the constrain $BLR \leq \theta$. The approach here discussed explores the space of solutions by a using random method. Let γ be the maximum value of the objective function for the optimal couple $(n_c, k_c)^{opt}$, such that:

$$\mathbb{P}_{\Omega_{(n_c,k_c)}} \left\{ \mathbf{O}_T((n_c, k_c)) \geq \gamma \right\} = \mathbb{E}_{\Omega_{(n_c,k_c)}} \left\{ \mathcal{I}_{\mathbf{O}_T((n_c,k_c))} \geq \gamma \right\}. \tag{4.8}$$

The weights ω_c that maximize the probability of the event $\mathbf{O}_T((n_c, k_c)) \geq \gamma$ in (4.8) must be estimated. This problem is equivalent to a Stochastic Node Network (SNN) formulation with a set of nodes $\{1, ...U\}$ and a set of node characteristics $\{1, ...C\}$ distributed according to (4.1). The objective is to assign the node characteristics while maximizing the objective function. Assuming that the nodes of the network draw (n_c, k_c) independently from each other and that ω_c is the probability for the couple c to be drawn, the vector $\mathbf{\Omega} = \{\omega_1, ...\omega_c\}$ represents the PMF distribution from which any node draws its coding scheme. This kind of SNN problems can be solved as follows: the steps shown in Algorithm 1 describe the solution here implemented and later evaluated. The $(1-\alpha)$-quantile $\hat{\gamma}^{(t)}$ of the estimated PMF $\hat{\Omega}^{(t)}_{(n_c,k_c)}$ can

Algorithm 1 Algorithm for the Optimal PMF

generate the initial distribution $\mathbf{\Omega}_0 = [\hat{\omega}_1^{(0)}, ...\hat{\omega}_c^{(0)}]$ *with entries* $\hat{\omega}_c^{(0)} = \frac{1}{C}$*;*

while $D_{KL}\left(\hat{\Omega}^{(t)}_{(n_c,k_c)}, \hat{\Omega}^{(t-1)}_{(n_c,k_c)}\right) \geq \epsilon$ **do**

generate S *random vectors* $\{(n_c, k_c)_s \cdots (n_c, k_c)_s\}$; $s = 1 \cdots S$ *of the* u *coding schemes in use by the* u *users such that* $\{(n_c, k_c)_s^1 \cdots (n_c, k_c)_s^u\} \sim \hat{\Omega}^{(t-1)}_{(n_c,k_c)}$*;*

evaluate the (1-α*)-quantile of the sample coding schemes, for which the objective function is greater than* $\hat{\gamma}^{(t-1)}$ *;*

update the estimation $\hat{\Omega}^{(t-1)}_{(n_c,k_c)}$ *to* $\hat{\Omega}^{(t)}_{(n_c,k_c)}((n_c, k_c))$ *through the samples that belong to the (1-*α*)-quantile;*

evaluate the KL-Distance $\left(\hat{\Omega}^{(t)}_{(n_c,k_c)} - \hat{\Omega}^{(t-1)}_{(n_c,k_c)}\right)$*;*

t=t+1;

end while

be evaluated by using the following formulation:

$$\hat{\gamma} = \max\left\{o \mid \mathbb{P}_{\hat{\Omega}_{(n_c,k_c)}}(\mathbf{O}_T((n_c,k_c)) \geq o) \geq \alpha\right\}. \tag{4.9}$$

The update of the estimation $\hat{\Omega}^{(t-1)}_{(n_c,k_c)}$ can be performed as:

$$\hat{\omega}_c^{(t)} = \frac{\sum_{s=1}^{S} \mathcal{I}_{\{\mathbf{O}_T((n_c,k_c)_s) \geq \hat{\gamma}^{(t)}\}} \mathcal{I}_{\{\omega_c\}}}{\sum_{s=1}^{S} \mathcal{I}_{\{\mathbf{O}_T((n_c,k_c)_s) \geq \hat{\gamma}^{(t)}\}}}. \tag{4.10}$$

The Kullback-Leibler distance (KL-Distance) between $\hat{\Omega}^{(t)}_{(n_c,k_c)}$ and $\hat{\Omega}^{(t-1)}_{(n_c,k_c)}$ can be evaluated as:

$$D_{KL}(\hat{\Omega}^{(t)}_{(n_c,k_c)}, \hat{\Omega}^{(t-1)}_{(n_c,k_c)}) = \sum_{U} \hat{\omega}_c^{(t-1)} \log\left(\frac{\hat{\omega}_c^{(t)}}{\hat{\omega}_c^{(t-1)}}\right). \tag{4.11}$$

Eq. (4.11) shows that the weights of PMF in (4.1) at time t are the occurrences of the coding scheme of the samples belonging to the $(1-\alpha)$-quantile. To amplify the selection of the weights of the PMF distribution in (4.1), the entries of the estimated $\hat{\Omega}^{(t)}_{(n,k_c)}$ are modulated as:

$$\hat{\omega}_c^{(t)} = \begin{cases} \hat{\omega}_c^{(t-1)} \left(1 + \dfrac{T((n,k_c)_c)}{\sum_{c=1}^{C} T((n,k_c)_c)}\right) & , c \in S_{best} \\ (1 - \sum_{c=1}^{C} \hat{\omega}_c^{(t-1)})\hat{\omega}_c^{(t-1)} & , c \notin S_{best},\ c' \in S_{best} \end{cases} \tag{4.12}$$

In (4.12), S_{best} is the set of coding scheme samples belonging to the $(1-\alpha)$-quantile. The first row shows how the weights of the coding schemes belonging to S_{best} are increased proportionally with the increment of throughput given by the use of the sub-optimal c coding schemes, iteration by iteration. Instead, the second row shows how the other weights are proportionally decreased, in a way that the new PMF does not change abruptly w.r.t. the previous one.

$w_d\,[ms]$	k_c	n_c	G_{est}	S_n	BLR	T
1000	1	1	0.111	10	0.01	0.099
1000	1	3	0.125	100	0.0	0.1
1000	5	10	0.125	100	0.2	0.5

Table 4.2: Sample data from an observation window

4.1.2.3 Empirically built Load Control strategy

In this Section, an empirically built technique is presented; then, it is compared, in Section 4.1.3, to the one previously described in Section 4.1.2.2. The idea is to find a control algorithm able to perform, at least, as the aforementioned one, while removing the need for a priori data. Let us assume that u users access the RA channel, each using a different linear code (n_c, k_c) extracted from a code set C, so that the average value $(\overline{n_c}, \overline{k_c})$ can be estimated.

In order to control BLR, a feedback loop is again exploited: an observer-like technique based on a window-based algorithm is the core of the control strategy. The frame decoding process provides information to the observer about the coding scheme, throughput and BLR per RCST. By taking advantage of these data, the statistics so computed can be used to control the behaviour of the system, and so its performance level. Table 4.2 shows the structure of the typical observation window. In the first column, the window duration $w_d = (t_e - t_s)$ is shown, where t_s and t_e are the timestamps representing the start and the end times of the observation window. The remaining columns show the transmitted information length k_c, the code length n_c, the estimated load G_{est} and the number of collected samples S_n within the window for a given coding scheme (i.e., the number of RCSTs that have transmitted during w_d using that linear code). The last two columns show the average values of BLR and T of the linear code in the first column. Note that a single window can be viewed as a set of tuples of cardinality equal to the number of different linear codes that each RCST has randomly drawn within w_d [ms]. The control algorithm periodically analyses the last available window data, with a period of LC_p [s], in order to estimate the average $(\overline{n_c}, \overline{k_c})$ and BLR values. The controller shifts the $(\overline{n_c}, \overline{k_c})$ values by changing the weights

of PMF $\Omega_{(n_c,k_c)}$, in order to improve the throughput in the upcoming window.

The main idea behind this control algorithm is to target a given BLR in the shortest possible time. BLR can be controlled by increasing the probability of drawing a code that fosters a BLR value closer to the target value. For instance, referring to Table 4.2 and assuming $BLR_{target} = 0.18$, BLR can be forced to a value close to the one in the last row by increasing the probability to draw the linear code $(n_c, k_c) = (10, 5)$.

The controller periodically acts as described in Figure 4.2: it receives as inputs G and the last observation window. These data are used to estimate BLR, $(\overline{n_c}, \overline{k_c})$ and T.

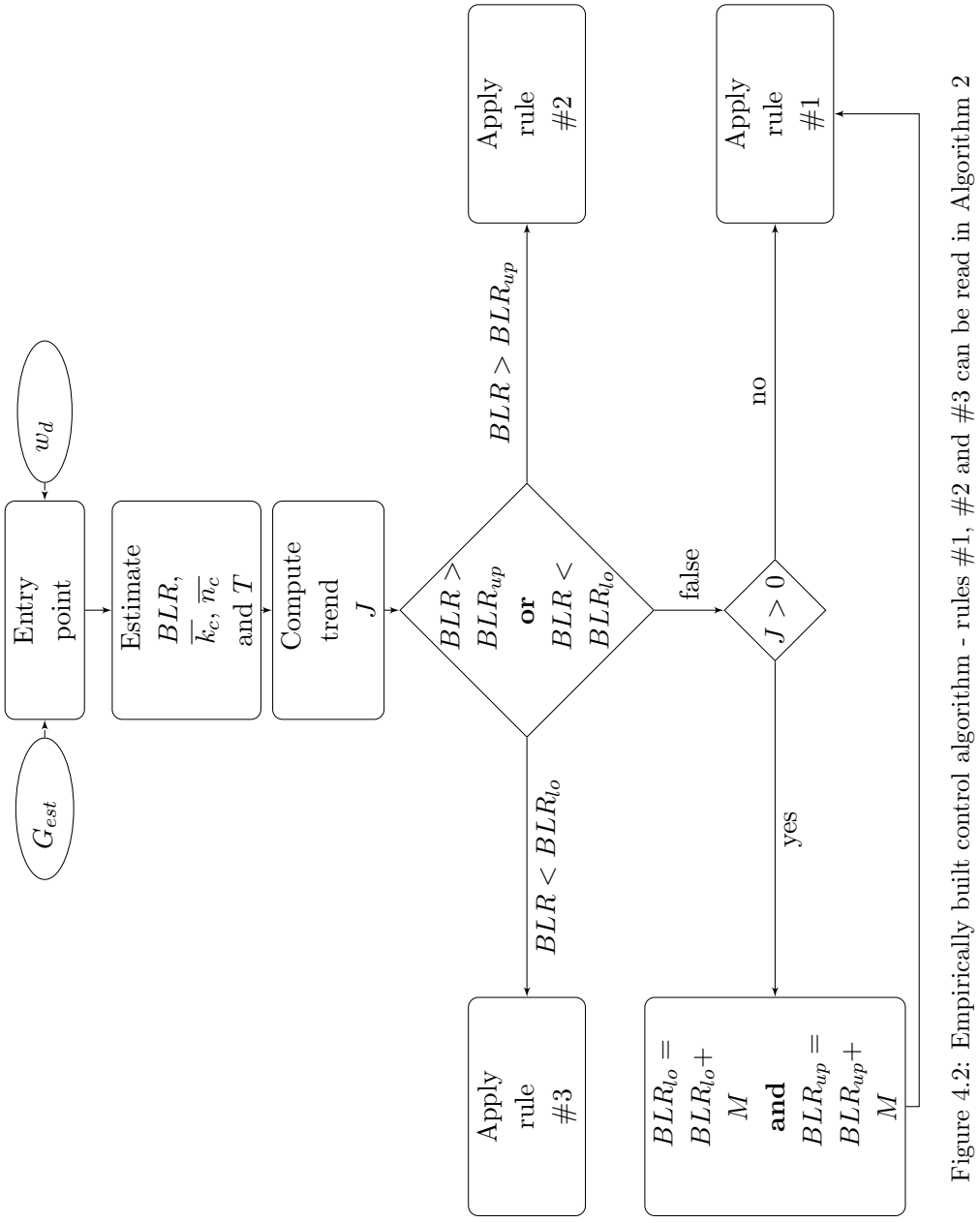


Figure 4.2: Empirically built control algorithm - rules #1, #2 and #3 can be read in Algorithm 2

Let $T^{(i)}$ and $BLR^{(i)}$ be the values at time i and T_{max} the maximum experienced throughput at the current load level. The so-called *trend* J is defined as:

$$\mathcal{J} = sgn((T_{max} - T^{(i)}))| \ (T_{max} - T^{(i)}) - (BLR^{(i)} - BLR^{(i-1)})| \ . \quad (4.13)$$

This formulation ensures that J is strictly positive only if T is increasing and its value depends on the difference between $(T^{(i)} - T^{(i-1)})$ and $(BLR^{(i)} - BLR^{(i-1)})$. The trend is a metric used to evaluate the system behaviour: if $J > 0$, then the system is performing better than in the previous window. BLR_{target} is upper and lower bounded by the values BLR_{up} and BLR_{lo}, respectively, where $BLR_{up} - BLR_{lo} > 0$. The target BLR is defined as $BLR_{target} = (BLR_{up} - BLR_{lo})/2$. The distance d between the current BLR and BLR_{target} is defined as $d = | \ BLR_{target} - BLR \ | \ .$ The parameter d is used to compute the coefficient $g = 0.5d$. The control algorithm adds g to selected ω_c and accordingly reduces the other weights, so that the increment of ω_c grows linearly with d. g is upper bounded, $g \leq 0.5$, which in turns translates into a maximum increment of 50% of the probability of a given (n_c, k_c) to be drawn. A larger g means that a faster change in the system behaviour is desired; instead, a reduced value means the algorithm is performing fine tuning because $BLR \approx BLR_{target}$.

To recap, the algorithm estimates BLR and T by using the last available observation window; then J is computed. Given the current G, when $J > 0$ and $BLR_{lo} \leq BLR \leq BLR_{up}$, one of the following conditions was verified in the observation window at time $(i-1)$:

- $(T^{(i-1)} - T^{(i-2)}) > (BLR^{(i-1)} - BLR^{(i-2)})$: the throughput grew faster than BLR;
- $(BLR^{(i-2)} - BLR^{(i-1)}) > (T^{(i-1)} - T^{(i-2)})$: BLR decreased faster than throughput;
- $(T^{(i-1)} - T^{(i-2)}) > 0$ and $(BLR^{(i-1)} - BLR^{(i-2)}) < 0$: the throughput grew and BLR decreased.

If $J > 0$, the following is done: $BLR_{lo} = BLR_{lo} + M$ and $BLR_{up} = BLR_{up} + M$, where M is an empirically found tuning factor reported in Table 4.4. The rationale behind the latter choice is the following: if the system is behaving

Parameter	Value
RA channel bandwidth	8 [MHz]
Modulation	Quadrature Phase-Shift Keying (QPSK)
Physical layer FEC rate	1/3
Frame duration	26 [ms]
Time-slots n per frame	100
Nominal RTT	520 [ms]

Table 4.3: Simulation parameters of the case study

better than in the previous observation window, we explore the possibility of further increasing the performance level by allowing a higher BLR. A higher BLR can be counteracted by a different choice of a linear code (n_c, k_c): if a linear code able to correct the erasures due to a higher BLR can be found, then it is likely that the throughput will increase, too; otherwise, the system stabilises its current performance level by finely tuning the choice of the linear codes in the current BLR range. In Algorithm 2 the aforementioned steps are described in pseudocode, implementing the parameters in Table 4.4.

4.1.3 Numerical Results

The previous Section has described the mathematical framework behind GE-CRDSA. The empirically built load control algorithm allows each RCST to randomly choose the number of information and parity packets, while the GCE-based algorithm relies on a heuristic to achieve the same, in order to improve the performance level of the system. In this Section it is shown how a given PMF, and consequently the choice of (n_c, k_c), affects T and BLR at different G values. An ad-hoc simulator has been developed to test the two strategies: the GCE-based and the empirically built one. The system and simulation parameters of the satellite scenario under consideration are summarized in Table 4.3.

The simulation results presented below are based on the use of two different code sets: firstly, the two strategies are compared in Section 4.1.3.1 by forcing them to use the same set of linear codes (as in Table 4.1). Then, in

Algorithm 2 Empirically built Load Control algorithm

if $\{BLR_{lo} \leq BLR \leq BLR_{up}\}$ **then**
  (RULE #1)
  $\omega_j = \omega_j + g;\ j = (n_c, k_c - 1)$
  $\omega_i = \omega_i + g;\ i = (\frac{\overline{n_c}}{\overline{k_c}}(k_c + 1), k_c + 1)$
  $(\overline{n}, \overline{k}) \approx (n_c, k_c)$
  subject to: $2k_c \leq n_c \leq 4k_c$
else if $\{BLR \geq BLR_{up}\}$ **then**
  (RULE #2)
  $\omega_j = \omega_j + g;\ j = (k_c, n_c)$
  $k_c = \min\{(\overline{k_c} - (\overline{k_c} - d\, k_{min}), k_{min})\}$
  if $k_c \neq k_{min}$ **then**
    $n_c = (\overline{n_c}/\overline{k_c})\, k_c$
  else
    $n_c = \max(n_c - 1, n_{min})$
  end if
  subject to:
  if $G_{est} \leq G_{peak}$ **then**
    $3k_c \leq n_c \leq 4k_c$
  else
    $2k_c \leq n_c \leq 4k_c$
  end if
else if $\{BLR \leq BLR_{up}\}$ **then**
  (RULE #3)
  $k_c = \min(k_c + 1, k_{max})$
  $n_c = k_c\, (\overline{n_c}/\overline{k_c})$
  subject to: $2k_c \leq n_c \leq 4k_c$
end if

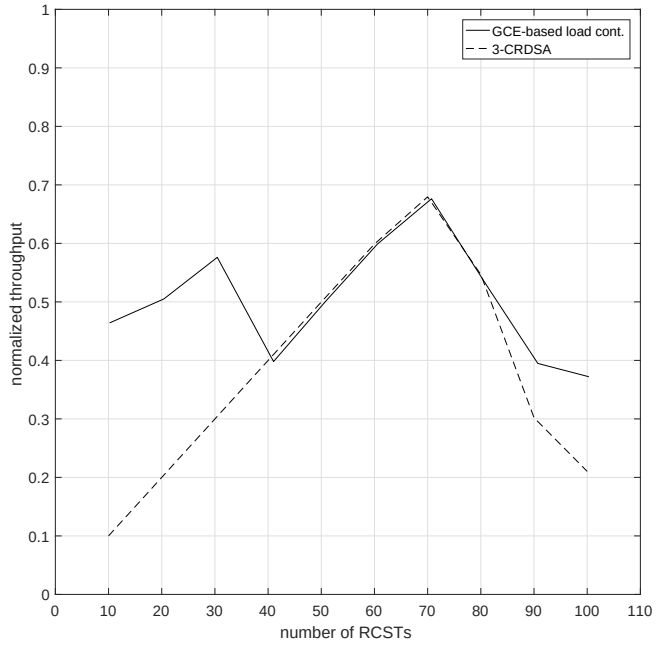
Section 4.1.3.2, the code set is extended to include also *non optimal* linear codes, i.e., linear codes different from the ones in Table 4.1: the extended code set respects the constraints $k_{min} \leq k_c \leq k_{max}$ and $n_{min} \leq n_c \leq n_{max}$, where the upper and lower bounds can be read in Table 4.4.

Description	Parameter	Value
Min info size	k_{min}	1
Max info size	k_{max}	10
Min code length	n_{min}	1
Max code length	n_{max}	15
LC chosen k	k_c	$[k_{min}, k_{max}]$
LC chosen n	n_c	$[n_{min}, n_{max}]$
G peak threshold	G_{peak}	0.80
P_L tuning factor	M	0.005
Initial $P_{L(up)}$ value	$P_{L(up)}$	0.03
Initial $P_{L(lo)}$ value	$P_{L(lo)}$	0.01
Ω_i at $t = 0$	Ω_i^0	$1/length(C)$
Control interval	LC_p	2 [nominal RTT]
Window duration	w_d	1 $[LC_p]$

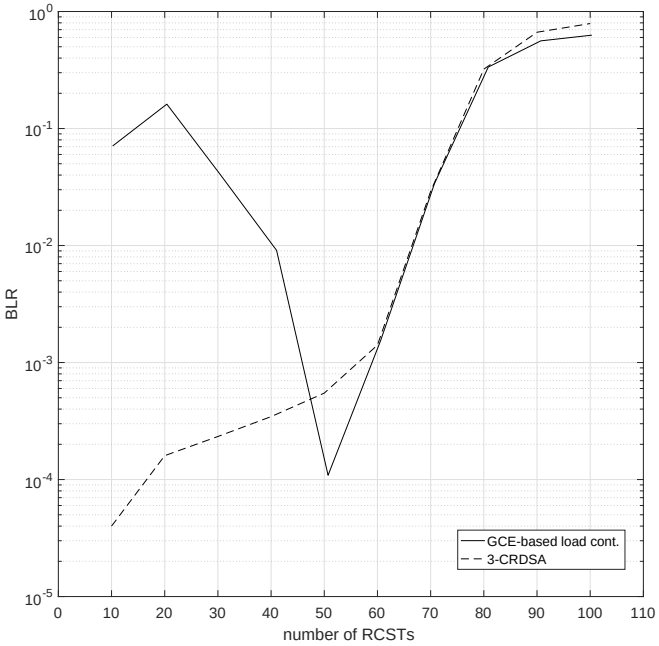
Table 4.4: Simulation and tuning parameters of the empirically built load control algorithm.

4.1.3.1 Limited code set

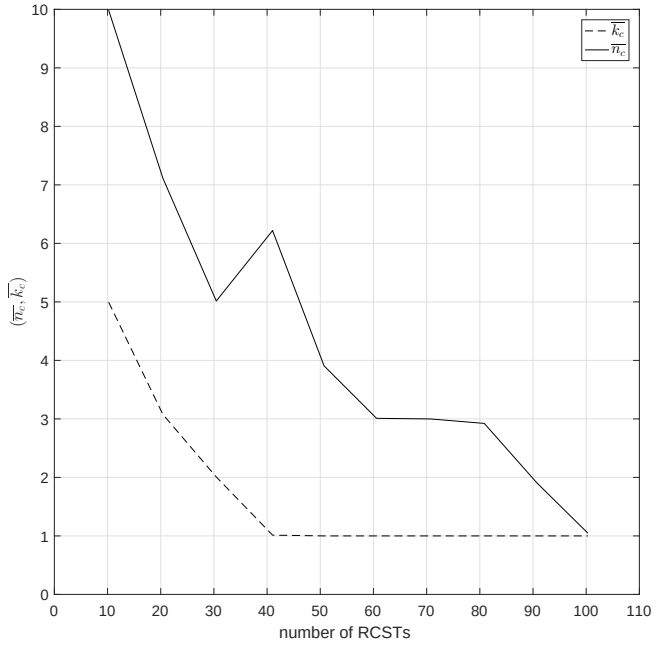
Figures 4.3 and 4.4 show the simulation results when applying the GCE-based control strategy and the empirically built one, respectively, for an increasing number of RCSTs. The GCE-based strategy offers throughput gains mainly at low loads, visible in Figure 4.3a, and starting from $G = \frac{40RCSTs}{n}$, the offered throughput is equal to the one offered by 3-CRDSA. On the other hand, BLR is higher than 3-CRDSA one, on average, as expected, and is visible in Figure 4.3b; finally, $(\overline{n_c}, \overline{k_c})$ is visible in Figure 4.3c at each G under consideration. The chosen linear codes by the GCE-based control strategy at each G can be read in Table 4.1.



(a) Throughput of the GCE-based load control algorithm vs 3-CRDSA throughput.



(b) BLR of the GCE-based load control algorithm vs 3-CRDSA throughput.



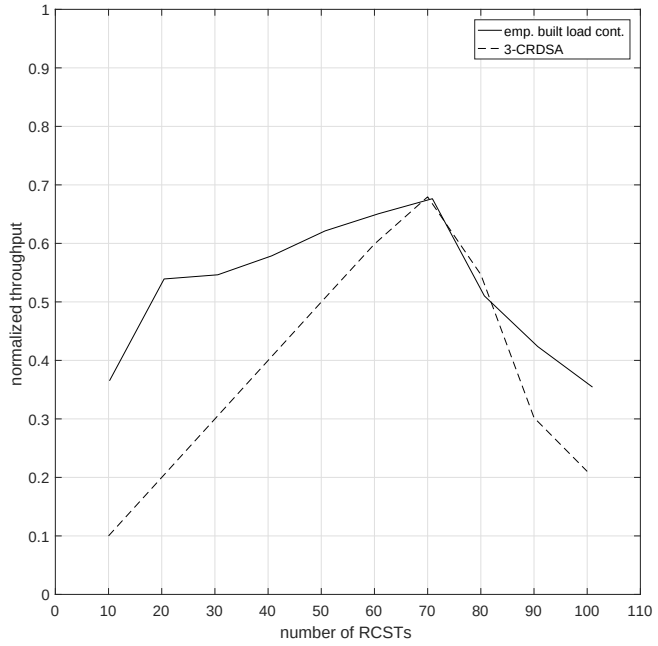
(c) $(\overline{n_c}, \overline{k_c})$ chosen by the GCE-based load control algorithm. CRDSA is equivalent to a fixed $(\overline{n_c}, \overline{k_c}) = (3, 1)$.

Figure 4.3: Performance of the GCE-based control strategy, limited linear code set C as in Table 4.1.

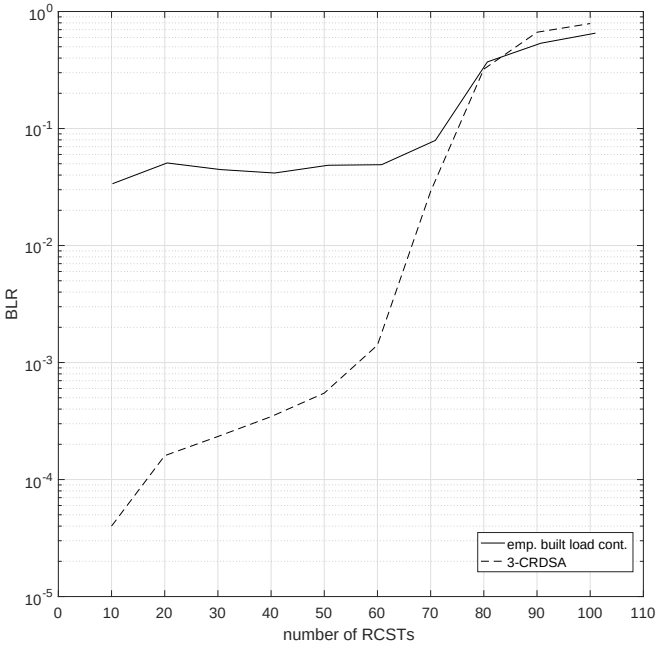
Instead, the empirically built control strategy is able to offer throughput gains up to $G \approx \frac{70RCSTs}{n}$, outperforming the GCE-based strategy, as in Figure 4.4a. The experienced BLR is higher, as in Figure 4.4b, and $(\overline{n_c}, \overline{k_c})$ is visible in Figure 4.4c at each G under consideration. Finally, the chosen linear codes are readable in Table 4.5: each row is presented as

$$\omega_c^1(k_c^1, n_c^1) + \omega_c^2(k_c^2, n_c^2) + ... \omega_c^p(k_c^p, n_c^p),\ 0.02 \leq \omega_c^i < \omega_c^{i-1}; \tag{4.14}$$

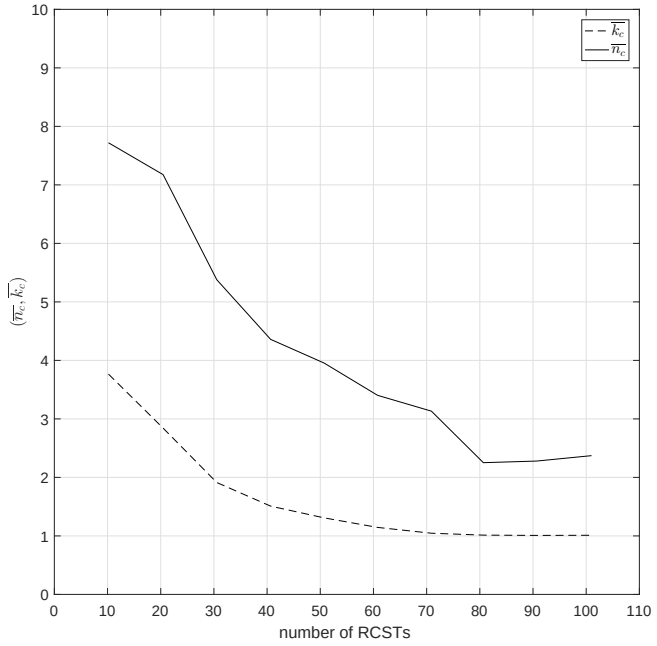
because of this, it may occur that $\sum_i \omega_c^i(k_c^i, n_c^i) \leq 1$ for space reasons in Tables 4.5, 4.6 and 4.7.



(a) Throughput of the empirically built based control algorithm vs 3-CRDSA throughput.



(b) BLR of the empirically built control algorithm vs 3-CRDSA throughput.



(c) $(\overline{n_c}, \overline{k_c})$ chosen by the empirically built control algorithm. CRDSA is equivalent to a fixed $(\overline{n_c}, \overline{k_c}) = (3, 1)$.

Figure 4.4: Performance of the empirically built control strategy, limited linear code set C as in Table 4.1.

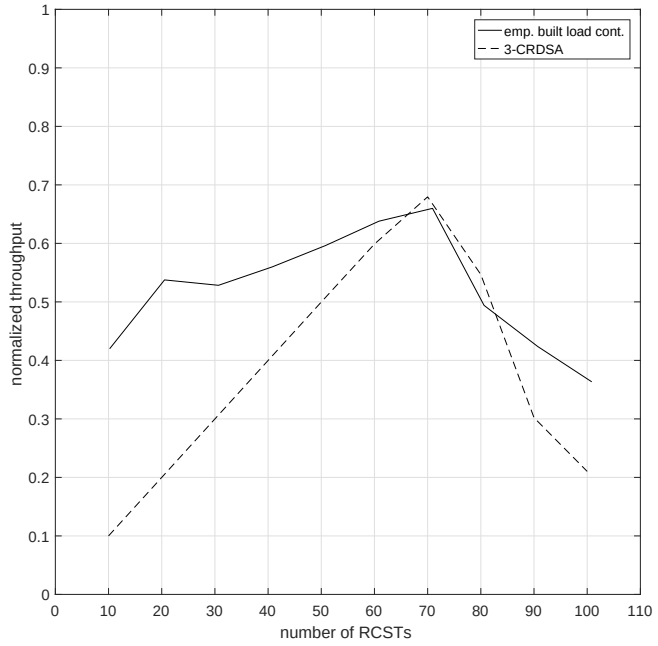
4.1.3.2 Extended code set

Figures 4.5 and 4.6 show the simulation results when applying the empirically built control strategy for an increasing and a decreasing traffic load, respectively. The GCE-based strategy would show here the same performance level as in the previous case, because it does not rely on linear codes whose statistics are not available prior to use. Here, the code set in use respects the constraints $k_{min} \leq k_c \leq k_{max}$ and $n_{min} \leq n_c \leq n_{max}$, where the upper and lower bounds can be read in Table 4.4. When enlarging the code set C each RCST can use, some throughput gain is still possible w.r.t. the code set in

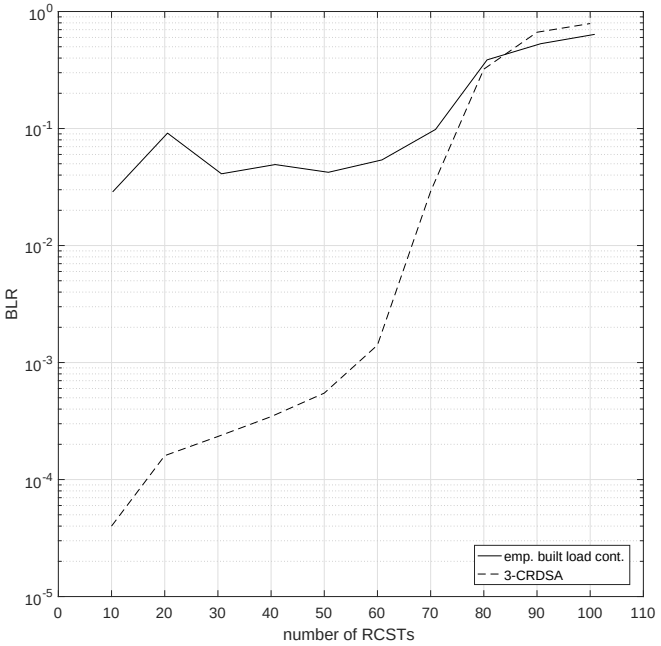
G	**Combination of the chosen linear codes:** $\omega_c^i\,(k_c^i/n_c^i)$
0.1	0.29*(3/9) + 0.06*(5/10) + 0.05*(4/8)
0.2	0.29*(2/7) + 0.25*(2/6) + 0.11*(3/9)+ 0.08*(4/8) + 0.06*(5/10)
0.3	0.31*(1/4) + 0.17*(2/7) + 0.14*(1/3) + 0.12*(2/6) + 0.09*(3/6)
0.4	0.45*(1/3) + 0.21*(1/4) + 0.11*(2/6) + 0.06*(3/6) + 0.06*(2/7)
0.5	0.60*(1/3) + 0.18*(2/6) + 0.16*(1/4)
0.6	0.78*(1/3) + 0.08*(1/4) + 0.07*(2/6) + 0.03*(2/4)
0.7	0.91*(1/3) + 0.04*(1/4)
0.8	0.82*(1/2) + 0.16*(1/3)
0.9	0.83*(1/2) + 0.16*(1/3)
1.0	0.80*(1/2) + 0.18*(1/3) + 0.02*(1/4)

Table 4.5: Chosen (k_c, n_c) at each G under consideration by the empirically built control strategy, limited linear code set C as in Table 4.1. Note that each line of the table shows only the most used codes for space reasons.

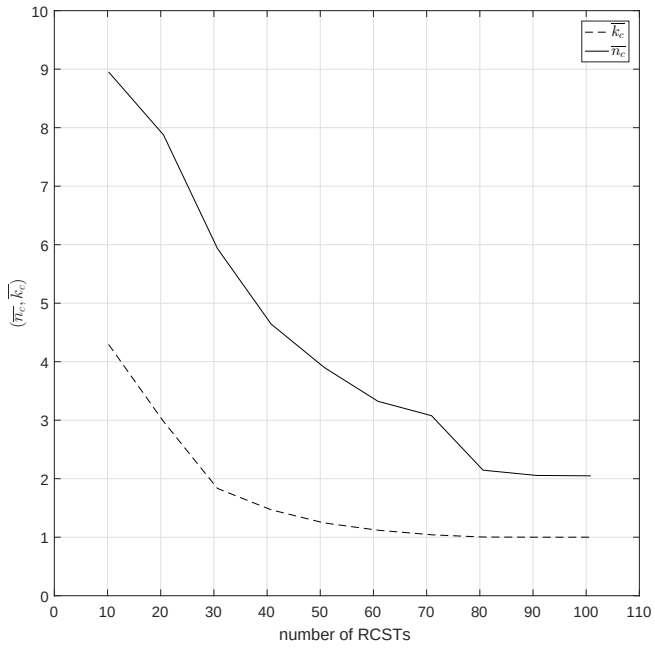
Section 4.1.3.1: for instance, in Figure 4.5a, at $G = \frac{10RCSTs}{n}$, the throughput value is slightly larger than the one visible in Figure 4.4a. Instead, in Figure 4.6a, at the same $G = \frac{10RCSTs}{n}$, T is slightly lower: in fact, the empirically built control strategy choices the linear codes to be used at each G by exploring the code set space, and different iterations (i.e., for instance, an increasing or a decreasing traffic profile) can lead to slightly different results. Finally, Tables 4.6 and 4.7 show the chosen linear codes at each load level for an increasing and a decreasing traffic profile, respectively.



(a) Throughput of the empirically built control algorithm vs 3-CRDSA throughput.



(b) BLR of the empirically built control algorithm vs 3-CRDSA throughput.

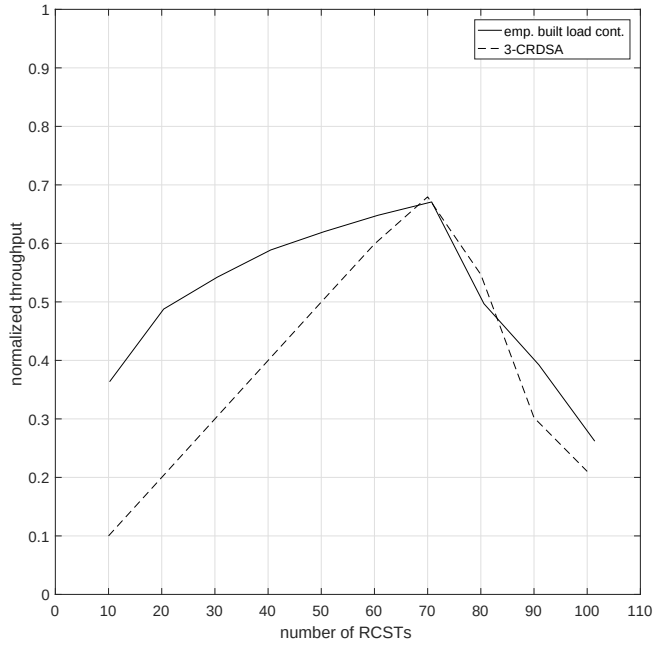


(c) $(\overline{n_c}, \overline{k_c})$ chosen by the empirically built control algorithm. CRDSA is equivalent to a fixed $(\overline{n_c}, \overline{k_c}) = (3, 1)$.

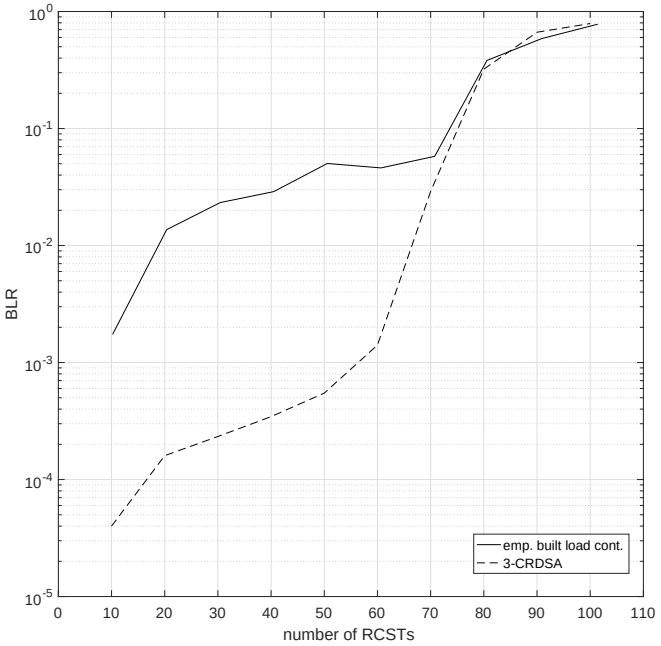
Figure 4.5: Performance of the empirically built control strategy for increasing G, extended linear code set C as in Table 4.4.

G	**Combination of the chosen linear codes:** $\omega_c^i\,(k_c^i/n_c^i)$
0.1	0.14*(4/12) + 0.078*(5/10) + 0.06*(4/8) + 0.05*(5/11)
0.2	0.25*(2/7) + 0.25*(2/6) + 0.23*(3/9) + 0.05*(4/12) + 0.04*(5/11)
0.3	0.51*(1/4) + 0.19*(3/9) + 0.074*(2/7) + 0.06*(2/8) + 0.048*(3/6)
0.4	0.38*(1/3) + 0.3*(1/4) + 0.065*(3/9) + 0.064*(2/6) + 0.055*(2/8)
0.5	0.65*(1/3) + 0.15*(1/4) + 0.095*(2/6) + 0.036*(2/8)
0.6	0.83*(1/3) + 0.058*(2/6) + 0.037*(1/4)
0.7	0.93*(1/3) + 0.017*(2/4) + 0.015*(1/2)
0.8	0.87*(1/2) + 0.12*(1/3)
0.9	0.94*(1/2) + 0.056*(1/3)
1.0	0.95*(1/2) + 0.05*(1/3)

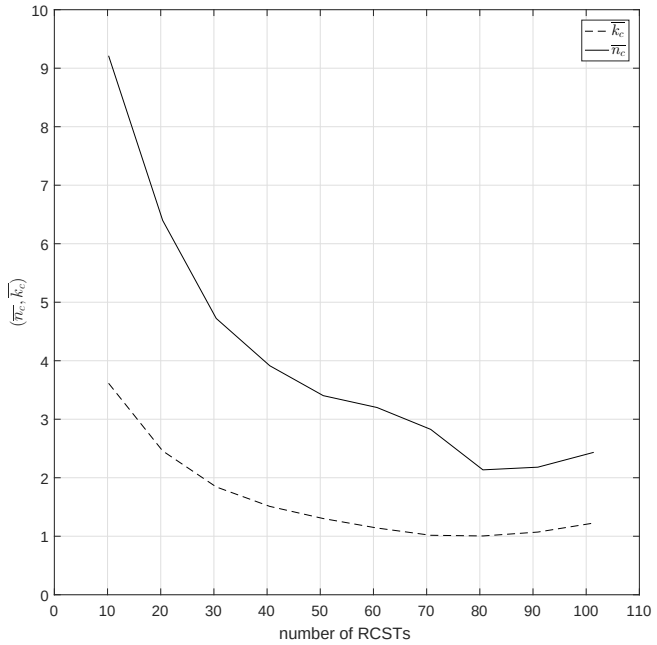
Table 4.6: Chosen (k_c, n_c) for increasing G under consideration by the empirically built control strategy, extended linear code set C as in Table 4.4. Note that each line of the table shows only the most used codes for space reasons.



(a) Throughput of the empirically built control algorithm vs 3-CRDSA throughput.



(b) BLR of the empirically built control algorithm vs 3-CRDSA throughput.



(c) $(\overline{n_c}, \overline{k_c})$ chosen by the empirically built control algorithm. CRDSA is equivalent to a fixed $(\overline{n_c}, \overline{k_c}) = (3, 1)$.

Figure 4.6: Performance of the empirically built control strategy for decreasing G, extended linear code set C as in Table 4.4.

The empirically built control strategy is not limited by the need of any a priori information, so it is able to exploit any code sets: instead, the GCE-based one does not use linear codes whose statistical information are not preliminarily available. Thus, the empirically built control strategy offers larger flexibility and can also adapt to system configurations using a different number of time-slots per frame: in fact, as demonstrated in [46], changing the number of available time-slots shifts G^*, thus, if using the GCE-based control strategy, the a priori information should be updated to take into account the aforementioned variation. On the other hand, the a priori information allows the GCE-based strategy to react faster to any load variations, because the optimal choice of linear codes at each G is already known; instead, the

G	**Combination of the chosen linear codes:** $\omega_c^i\,(k_c^i/n_c^i)$
1.0	0.94*(1/2)
0.9	0.94*(1/2) + 0.04*(1/3)
0.8	0.90*(1/2) + 0.10*(1/3)
0.7	0.80*(1/3) + 0.19*(1/2)
0.6	0.76*(1/3) + 0.09*(1/2) + 0.07*(2/4)
0.5	0.57*(1/3) + 0.12*(2/5) + 0.12*(2/4) + 0.12*(1/2)
0.4	0.47*(1/3) + 0.19*(2/5) + 0.10*(2/4) + 0.07*(1/2)
0.3	0.28*(1/3) + 0.15*(2/4) + 0.14*(1/4) + 0.13*(3/6)
0.2	0.36*(3/8) + 0.13*(1/4) + 0.13*(3/6) + 0.07*(1/3)
0.1	0.39*(4/10) + 0.18*(5/13) + 0.12*(3/8) + 0.09*(2/7)

Table 4.7: Chosen (k_c, n_c) for decreasing G under consideration by the empirically built control strategy, extended linear code set C as in Table 4.4. Note that each line of the table shows only the most used codes for space reasons.

empirically built one has no memory of its previous choices at a given G, thus the sub-optimal choice of linear codes must be find by re-exploring the code set space, and different iterations can lead to slightly different results. To prove the latter, Tables 4.6 and 4.7 show PMFs of selected combinations of linear codes at each considered G, when the traffic load is increasing and decreasing, respectively. If carefully read, the two Tables show that the final combinations of chosen linear codes is different at the same G: if a load level is reached from a lower/higher one, the aggregated throughput achievable by using the empirically built algorithm is approximately comparable (i.e. $\pm 5\%$), but RCSTs may rely on different choices of linear codes. In fact, at runtime, the performance level achievable by using the last choice of linear codes at the previous load level is used as reference to choose the new direction of exploration of the code set space, in order to find a new choice which may offer some throughput gain w.r.t. the current experienced T.

4.1.3.3 Convergence time

In this Section, the time needed by the empirical built controller to converge to the desired performance level is analysed. The desired performance level is defined as the one that ensures a selection of the linear codes that provides a minimum throughput at least equal to 90% of the value achievable if using the best selection of linear codes (per load level); for instance, the best selection can be read, for instance, in Table 4.6. The GCE-based control strategy immediately converges to the best solution per load level (in Table 4.1), thus it is ignored in this part.

Figure 4.7 shows the PMF of the convergence time [control intervals] needed to reach the desired performance level: the proposed controller converges to the desired performance level within 3 [control intervals] in the 98% of the simulations we ran. Once close to a sub-optimal selection of linear codes, it refines the said selection at the current load level to further improve the achievable throughput. A consideration is here in order: more samples can be collected in a larger w_d ($= LC_p$), thus the algorithm requires fewer (large) control intervals to reach the desired performance level; on the other hand, a smaller duration of the observation w_d reduces the number of collectable samples, reducing the amount of information gathered per LC_p, but RCSTs receive more frequently the updated weights ω_c: thus, more (small) control intervals are required to converge on the best selection of linear codes than using a larger w_d. The duration of the control interval LC_p should be adjusted according to how fast G varies: if G slowly varies in time, then the observation window w_d can be large (several RTTs); on the contrary, if G quickly varies in time, then the duration of w_d should be as small as possible ($w_d \geq RTT$).

4.1.4 Considerations and discussion

This Chapter has shown how relying on a linear code prior to transmission can improve the throughput, up to the twice of that obtained when using CRDSA, if the system is poorly loaded; the proposed algorithm works similarly to CSA, but a RCST can send more than one burst per RA block. Two different load control algorithms have been proposed and tested: the first algorithm relies

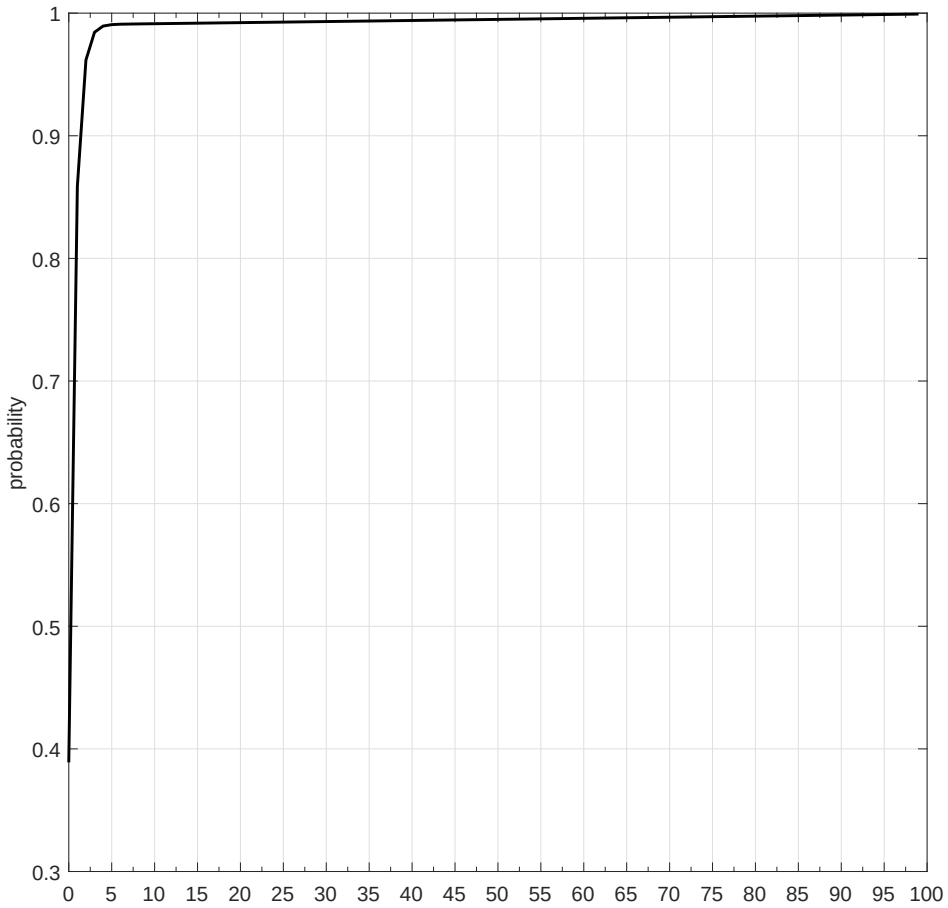


Figure 4.7: Probability Mass Function of the convergence time $[LC_p]$ needed by the proposed empirically built control strategy to reach a performance level close to the desired one

on a heuristic, the second one explores the space of the solutions at run-time. In both cases, the complexity of a Demand Assigned Multiple Access (DAMA)-like approach can be avoided, while achieving a good performance level. Since this study only accounts for colliding packets exhibiting the same Signal to Interference plus Noise Ratio (SNIR), further improvements in terms

of optimal system load and maximum achievable throughput can be obtained if also considering the power unbalance and the capture effects. However, it does not impact on the rationale behind the proposed scheme, which will reach a higher throughput value, as in unbalanced configurations [10].

The available pool of RA time-slots can be dynamically reconfigured when using a DVB-RCS2 system, in order to operate the system close to the G^* working point; the proposed empirically built strategy can obtain the same result without any frame length reconfiguration: the frame length reconfiguration strategy can be hardly applied to systems showing a rapidly varying traffic rate, as for instance in the case of M2M scenarios. A load control algorithm, instead, if properly tuned, provides the opportunity to track instantaneous G and BLR, and to react quickly, in order to exploit as much as possible the available resources on a wide range of offered loads. Another advantage of the empirically built control strategy is worth to be discussed: the proposed algorithm is resilient to errors in the load estimation process. In fact, GCE_{est}, which is received as input by the controller, is only used to discriminate if the current load level G is larger than G_{peak}. The choice of a local linear code over another is instead based the experienced BLR: thus, any estimation error has a very limited impact on the achievable throughput. Instead, if using the GCE-based control strategy, the choice of a linear code over another is solely based on G_{est}, thus any estimation error has a larger impact on the achievable throughput.

4.2 A new RA-DA hybrid MAC approach for DVB-RCS2

This Section describes an innovative satellite MAC scheme, aimed at efficiently addressing IoT/M2M traffic in DVB-RCS2 systems. This access scheme relies on the idea that DA schemes can be complemented, in a dynamic way, to RA schemes, when traffic spikes feed a satellite link. The available time-slots per superframe are here divided into two *pools*: one reserved for RA access , the other one reserved for DA access. The rationale is to control the offered load G on RA pool to keep BLR under a pre-defined threshold: here, BLR is assumed only due to collisions. A prototype implementation has been developed by using the NS-2 simulator [49], to verify the effectiveness of the proposed approach. The proposed architecture is promising and, in what follows, the main dynamics of this innovative hybrid scheme are highlighted.

4.2.1 Introduction

RA schemes in DVB-RCS2 standard are mainly utilized in presence of M2M applications: a RCST can initiate a data transmission in the upcoming RA block, without any delay. On the other hand, DA schemes are able to support a sustained traffic rate through a reservation mechanism. RCSTs send capacity requests to NCC when the data pending for transmission fill the MAC queue, in order to reserve DA time-slots for delivering data pending for transmission. Definitively, RA is able to guarantee an immediate beginning of the transmission, at the cost of collisions, whilst DA can guarantee data transmissions without any collisions, but at the cost of a set-up time.

To take advantage of the advantages of both approaches and to open to the idea of hybrid transmission schemes in DVB-RCS2 standard, the following study is presented. Please note that DVB-RCS2 standard opens to the possibility to allocate part of a superframe to RA transmission mode and the residual part to DA transmission mode. The architecture and the dynamic mechanisms are here designed and empirically tested: the main benefit is in the efficient management of traffic variations in a satellite network, allowing to control BLR and to guarantee a satisfying utilization of the available resources on a satellite return link. The prevalent hybrid access schemes available in

literature are described in Section 3.3.

4.2.2 Scenario under consideration

The scenario under consideration is composed by a set of remote sensors connected to a pool of RCSTs, sending M2M-like data to a remote server via satellite return link. A M2M traffic profile is taken into account; thus, a fraction of the overall return link capacity is managed with a RA protocol operating on a fraction of the superframe (a RA block). The RA block size depends on the number of RCSTs supporting sensor applications foreseen in the system, here assumed known a priori. The residual link capacity is eventually allocated in DA mode by the NCC to either sensor-based RCSTs or additional RCSTs relying on DA slots only: the latter are herein referred as full-DA RCSTs. In particular, the proposed access scheme allows a sensor-based RCST to exploit DA slots, if not already reserved by full-DA RCSTs, to efficiently manage temporary possible traffic spikes.

The main driver for the proposed access scheme leverages on the following consideration: the statistics than can be found in literature about BLR of RA schemes assume uniform or slow changing traffic loads. Instead, the vast majority of the real sensor applications produce data bursts over a timescale different from the channel competition interval, which is assumed correspondent to a superframe[3] in this work. Although the average statistics may show a good match with RA requirements in terms of BLR, the variance of G can be high. Under these conditions, the instantaneous BLR may be larger than an acceptable rate: the maximum tolerable value is here defined as BLR_{thr}. In this context, DA can act as a *bonus capacity*, aimed at keeping BLR lower than BLR_{thr}. Extra slots can be assigned to RCSTs with longer data queues, in order to reduce the competition in RA block. In addition, if a RCST shows a sustained load for several consecutive RA blocks, NCC can force the terminal to only rely on DA, working as a full-DA RCST. Assigning dedicated slots to greedy RCSTs aims at preserving the performance of M2M RCSTs, which relies on use of a RA scheme to deliver data. Thus, forcing greedy RCSTs to use DA access leads to a reduction of BLR in RA blocks, but at the cost of

[3]The minimum competition interval is equivalent to a RA block: we assume a single RA block per superframe, so the competition interval under consideration is the minimum one.

an increased latency due to DA reservation request (or access delay).

4.2.3 RA-DA protocol architecture and operations

Figure 4.8 sketches the architecture of the proposed MAC scheme. The physical queue of a RCST is controlled by a local scheduler, which decides if the next MAC burst must be transmitted in a RA slot or in a DA slot, the latter if an assignment has been notified by NCC for the upcoming superframe. Once a packet is scheduled for DA or RA transmission, a virtual allocator takes care of transmitting it in the next superframe by choosing a dedicated or a random time-slot, respectively. Thus, three operation modes are possible: *(i)* a RCST acts as a pure RA terminal; *(ii)* a terminal acts as a pure DA terminal; *(iii)* finally, a terminal can operate in a hybrid way, issuing a DA reservation request while exploiting RA access until the reservation response. NCC is in charge of the centralized decision of enabling/disabling pure RA and/or pure DA operation modes per RCST per superframe, as well as the tracking of the allocated and pending DA requests. NCC is also responsible for monitoring the load insisting on RA block, in order to keep BLR lower than BLR_{thr}. Retransmissions are not explicitly taken into account in this work, but a RCST that experiences large losses because of collisions can issue a DA reservation request to correctly deliver its data without further delay due to collisions.

A DA scheme requires some signalling between RCSTs and NCC. Herein, the VBDC algorithm[4] is assumed as reference. When acting as a DA RCST, the terminal issues a VBDC request to NCC, which replies in broadcast with a new Terminal Burst Time Plan (TBTP), satisfying the needed reservation in the upcoming superframes. DA set-up time, namely $T_{DA} = p$ [superframes], is computed starting from the reservation request to the reception of the new TBTP.

4.2.3.1 RA-DA basic operations

At the start-up, RCSTs are operating in RA mode: a single RA time-slot can be used per RA block. RCSTs monitor MAC queue size and, if the latter

[4]VBDC assignment scheme is described in Section 3.2.

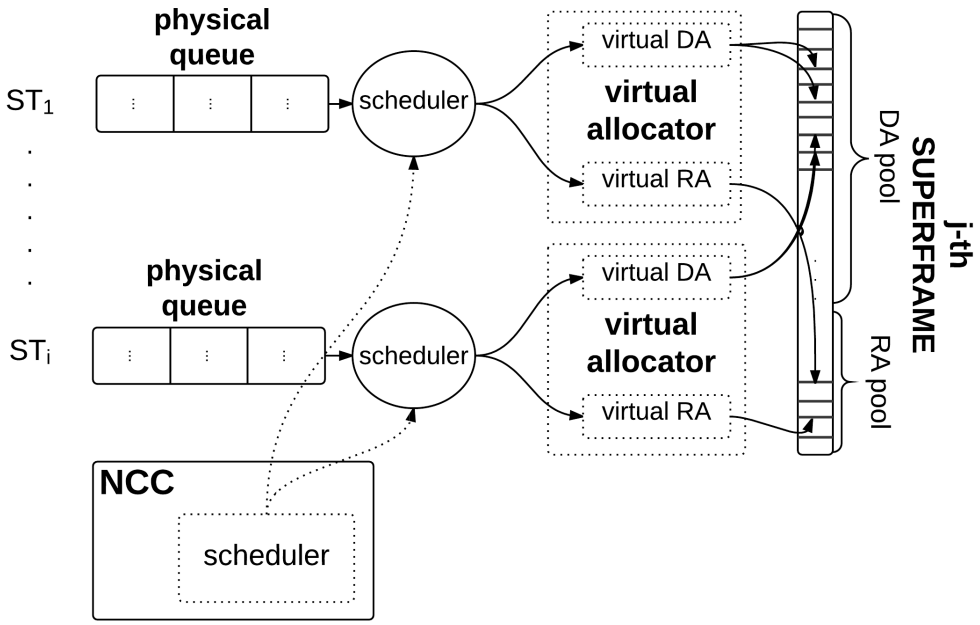


Figure 4.8: A sketch of the architecture of the RA-DA access scheme

exceeds the amount of bytes that can be served in p superframes just relying on RA slots, they issue a DA request to deliver the extra bytes. For instance, let us assume RA pool consisting of 20 time-slots and $p = 10$: if a single time-slot payload size is 38 [B] (for instance, as in Waveform 3[5] of DVB-RCS2 standard), a RCST with a queue length ≤ 0.38 [kB] will attempt to process data only relying on RA. However, if a high number of terminals, although respecting the above conditions, simultaneously transmit in RA block, the experienced collision rate can overcome BLR_{thr}. If this is the case, NCC forces a number of RCSTs to disable RA operation mode, forcing a pure DA operation mode. A fraction of RCSTs, ordered by the amount of pending DA reservation requests, will be forced to adopt DA only. In this way, some traffic is moved from RA pool to DA pool, lowering BLR. These terminals will be later re-admitted to RA mode, likely when their traffic will be again

[5] *Table A-1: Reference Waveforms for Linear Modulation Bursts*, Digital Video Broadcasting (DVB); Part 2: Lower Layers for Satellite standard, DVB Document A155-2, January 2013.

short-time bursty one. Therefore, the main goal algorithm is to control BLR in RA block. In case of temporary traffic spikes, the delay experienced by RCSTs forced to operate in DA mode increases, but this allows to operate in a region where $BLR \leq BLR_{thr}$.

NCC monitors the activity of RA-DA RCSTs by updating two logical binary arrays. The first array, namely RA_{record}, contains information about RCSTs operating in RA mode: *1* indicates that the use of the RA pool is possible, *0* viceversa. The second array, namely DA_{record}, stores the last DA requests map, in terms of time-slots. In addition, NCC tracks and sum DA requests of the last p consecutive superframes.

4.2.3.2 Operating mode: hybrid RCSTs

In this setup, RA allows a RCST to transmit a single MAC burst per superframe. A variable traffic profile is assumed, thus in every superframe two different situations are possible:

- no MAC burst are available for transmission;
- multiple MAC bursts are available for transmission.

The architecture proposed in this work exclusively leverages on RA operation mode if pending data can be fully transmitted in a time $t \leq T_{DA}$. The overall dynamic is presented through the example shown in Figure 4.9, where the activity of a single RA-DA hybrid terminal is summarized, assuming $p = 10$. The algorithm works as follows:

1. *superframe 1*: MAC queue contains 5 bursts pending for transmission. Considering a First In First Out (FIFO) policy, the first burst is sent in a RA time-slot and, since the remaining bursts can be sent within p, no DA reservation requests are issued;
2. *superframe 2*: two additional bursts are queued, increasing the total to 7; no reservation requests are issued because $p > 7$;
3. *superframe 3*: five additional bursts are queued, increasing the total to 11 bursts. The first is sent in a RA time-slot, and the remaining 10 can be addressed in T_{DA};

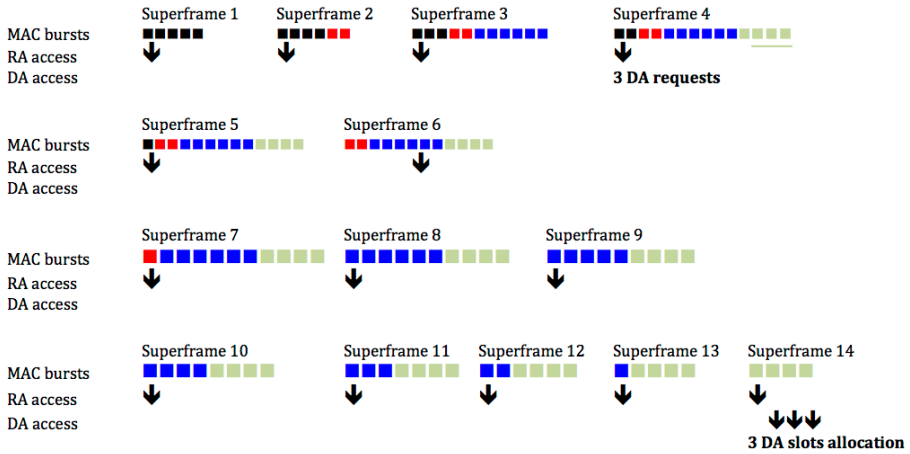


Figure 4.9: RA-DA combined assignment in an example scenario

4. *superframe 4*: four additional bursts are queued, increasing the total to 14 bursts. The first is sent in a RA time-slot, the next 10 will be sent in RA time-slots within T_{DA}, and the remaining 3 bursts trigger a DA reservation request. We recall that a reservation response (if any DA time-slots are available) will be received from NCC after T_{DA};
5. *superframes 5-13*: no new bursts queued. The first burst in the queue is sent in a RA time-slot. No further DA requests;
6. *superframe 14*: four bursts are still pending for transmission. The allocation of the three DA time-slots is now available (as requested by RCST in *superframe 4*). Therefore, the first burst is sent in a RA time-slot, while the residual three are sent in DA time-slots.

Afterwards, the MAC queue is empty. The arrival of new bursts will trigger the same procedure. In the previous example, without DA support, the terminal would have needed three more RA blocks to deliver the three bursts that have been processed in *superframe 14* relying on DA slots.

4.2.3.3 Operating mode: pure DA RCSTs

NCC uses the instantaneous RA load to estimate the instantaneous BLR, depending on the RA scheme in use. For instance, if SA is selected, the BLR

estimation can be analytically computed by using the following formulation: $T = G(1 - BLR) = Ge^{-G}$, where G is the instantaneous normalized offered load, T the instantaneous normalized throughput, and BLR the unknown to be determined as: $BLR = 1 - T/G = 1 - e^{-G}$. The instantaneous load estimation can be made by relying on the approach proposed in [47], which shows a large precision. This load estimation algorithm is thoroughly discussed in Section 4.1.2.1.

If $BLR > BLR_{thr}$, NCC can decide to disable RA operation mode for one or more RCSTs, to aim at respecting the constraint $BLR \leq BLR_{thr}$. The decision on which RCST will be inhibited to access the RA block (among those having RA option enabled, as tracked in the RA_{record} array) relies on the number of DA requests monitored in DA_{record} array. Then, NCC starts to block terminals from accessing RA block, by choosing the ones requiring the highest number of DA resources. BLR could exceed the target rate also when all RCSTs are relying on RA time-slots only. In this case, NCC can arbitrary select the terminals to temporary block from using RA operation mode (a random selection is envisaged). When a terminal is blocked from accessing RA block, it behaves as a full-DA terminal, sending DA resource requests for any queued burst. A blocked terminal can autonomously reactivate RA operation mode as soon as its queue length is less than Q_{RA} [superframes] (we assume that each item in the queue can be transmitted in a single superframe), a parameter that represents an upper bound under which the traffic of the station is considered eligible for RA operation mode. Considering that up to p bursts can be transmitted in RA operation mode within an allocation period, the threshold for re-enabling RA can be set as a fraction of p. The rationale is to carefully allow re-entering RA block only to RCSTs exhibiting low traffic rates, in order to minimize the impact on active RA RCSTs (i.e., having more RCSTs accessing the RA block increases BLR). In fact, prematurely re-admitting terminals may induce some sort of oscillations in the control loop, because of a continuous enabling/disabling process on RA option. The effect of disabling RA on a RCST transmission is summarized in the Figure 4.10. When a RCST receives a notification of RA disabling (as in superframe 6 of Figure 4.10), it immediately issues a DA request for delivering the queued MAC bursts. Within p, RCST sends only two additional DA requests (in

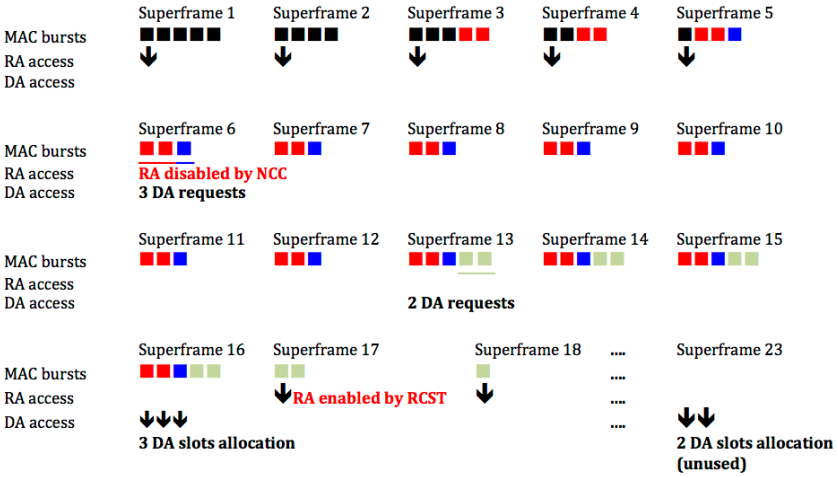


Figure 4.10: RA mode disabled by NCC and re-enabled later by RCST in an example scenario

superframe 13); assuming $Q_{RA} = p/2 = 5$ in the example in Figure 4.10, at superframe 17, RA operation mode is locally re-activated. In case of a higher number of DA requests (> 5) within p, the re-enabling of RA operation mode would be delayed until the condition of a queue length $\leq Q_{RA}$ is satisfied. To summarize, NCC can inhibit a RCST to use RA block in order to limit $BLR \leq BLR_{thr}$, but a RCST can automatically re-enable RA operation mode if its queue length $\leq Q_{RA}$.

4.2.4 Simulation scenario

In the following simulation results, UDP-based sensor sources are considered. The traffic generation relies on a negative exponential distribution for both message size and consecutive transmissions inter-time. For the sake of simplicity, the sources are assumed directly connected to a RCST. The performance metrics for evaluating the proposed algorithm are: delivery time of packets and offered load, in both RA and DA pools of each superframe.

4.2.4.1 Simulation setup

The simulation framework is NS-2-based. With respect to [21], several modifications are made to adapt the simulator to the current scenario:

- the NCC scheduler has been completely rewritten, in order to manage the varying pool of RCSTs relying on RA operation mode, also implementing the analytical load estimation model in [47];
- a simple iterative algorithm has been designed and implemented, in order to choose which RCSTs should disable RA operation mode, on a superframe basis;
- the architecture in Figure 4.8 has been implemented; more specifically, the virtual allocator and the scheduler have been developed from scratch.

The simulation platform considers 20 competing RCSTs, which transmit M2M data. The sensor generation pattern is regulated by the following random variables:

- *message size*: extracted from a negative exponential distribution with expected value μ;
- *interarrival time*: extracted from a negative exponential distribution with expected value λ.

Where not specified, the values of $\lambda = 0.5$ [s] and $\mu = 320$ [B] are used, resulting in an average G of one time-slot every five superframes. UDP-IP segmentation has been configured, in order to reproduce the correspondence between UDP size and available bytes per time-slot. SA access scheme is selected as RA protocol, since it represents the simplest approach. We recall that this is a preliminary work, and SA is selected in order to focus on the proposed hybrid approach feasibility in the worst case scenario and to preliminarily validate the achievable performance level, before assessing a further performance improvement in presence of more advanced RA schemes [10, 38, 46, 50]. Table 4.8 provides an overview of the simulator main settings.

Table 4.8: Main settings of the simulator

Parameter	Value
Number of active RCSTs	20
Superframe duration	98 [ms]
Time-slot size	376 [B]
Number of time-slot per superframe	64
Frames per superframe	1
RA pool size	20 [time-slots]
DA pool size	44 [time-slots]
Return link bandwidth	2 [Mbit/s]
RA protocol	SA

4.2.4.2 Simulation results

Extensive Montecarlo simulations have been ran. Each test has been performed with four different configurations, as follows:

1. *VBDC*: using DA-VBDC with 20 available time-slots, which represents the best solution for channel efficiency; it allows to compare the performances of DA and RA in a fair way;
2. *RA 20*: using a RA pool of 20 time-slots out of the 64 available in the system (up to 1 time-slot per RCST per superframe);
3. *RA 64*: using a RA pool of 64 time-slots out of 64; this represents a full RA scheme, where DA is not foreseen. Differently to the previous configurations, possible RCSTs leveraging on a parallel DA-only service cannot be supported;
4. *hybrid mode*: the hybrid algorithm proposed in this work, where the configuration used in the following simulations foresees a RA pool composed of 20 time-slots and a DA pool composed of 44 time-slots. Two different BLR_{thr} values have been considered: $BLR_{thr} = 0.1$ for *hybrid mode TH0.1* and $BLR_{thr} = 0.3$ for *hybrid mode TH0.3*, standing for the 10% and 30% usage of the RA allotment, respectively.

The first measured parameter is the average packet delivery delay, which is the time between a packet generation at the source and its reception at destination side, assumed beyond the satellite gateway. The results are shown in Figure 4.11: VBDC scheme shows the highest average delivery time, namely DT_{max}, because of the reservation delay. Its value is about 1.2 [s], which includes: one RTT for the capacity request-assignment message exchange, DA processing delay and the time required to actually transmit any data over the satellite return link. The pure RA scheme shows the lowest delivery time, namely DT_{min}, since the packets are directly sent in RA time-slots without any delay. Similar results are measured in the configurations *RA 20* and *RA 64*: this is because of the low aggregate G, which makes negligible the queuing delay, on average. The average value of the delivery time in the configurations *RA 20* and *RA 64*, about 0.4 [s], is higher than 0.25 [s] (the typical propagation delay of GEO satellites), and it derives from the framing delay needed to address the transmission of sensor messages in more than one superframe. When adopting the proposed RA-DA approach, the average delivery time is greater than DT_{min} and less than DT_{max}, as shown in the last two bars of Figure 4.11. Specifically, for $BLR_{thr} = 30\%$, the delivery time is 0.45 [s], close to DT_{min}: it means that only few MAC bursts are sent through DA time-slots because BLR_{thr} is rarely overcome. Setting $BLR_{thr} = 10\%$, instead, leads to a more frequent switch to a full DA operation mode, in which RCSTs behave as in DA-VBDC case. Therefore, the resulting average delay is inflated to 0.9 [s].

Varying BLR_{thr} has impact on the actual BLR value: Table 4.9 shows the sensitivity to this parameter and the empirically measured value: it results that $BLR > BLR_{thr}$. The latter phenomenon depends on the large delay on the feedback loop, i.e., the large delay due to the satellite link: the difference between BLR and BLR_{thr} would be lower if the propagation delay reduced. The simulated BLR values must be taken with a grain of salt, considering the poor performances of SA, already at medium loads. BLR is 0% when DA is used, and it can be fairly controlled adopting the proposed mechanisms, according to the target applications. The average delivery time reduces as BLR_{thr} increases: in fact, a larger BLR_{thr} means that the system is less likely to switch to DA operation mode, thus RCSTs are more likely to access

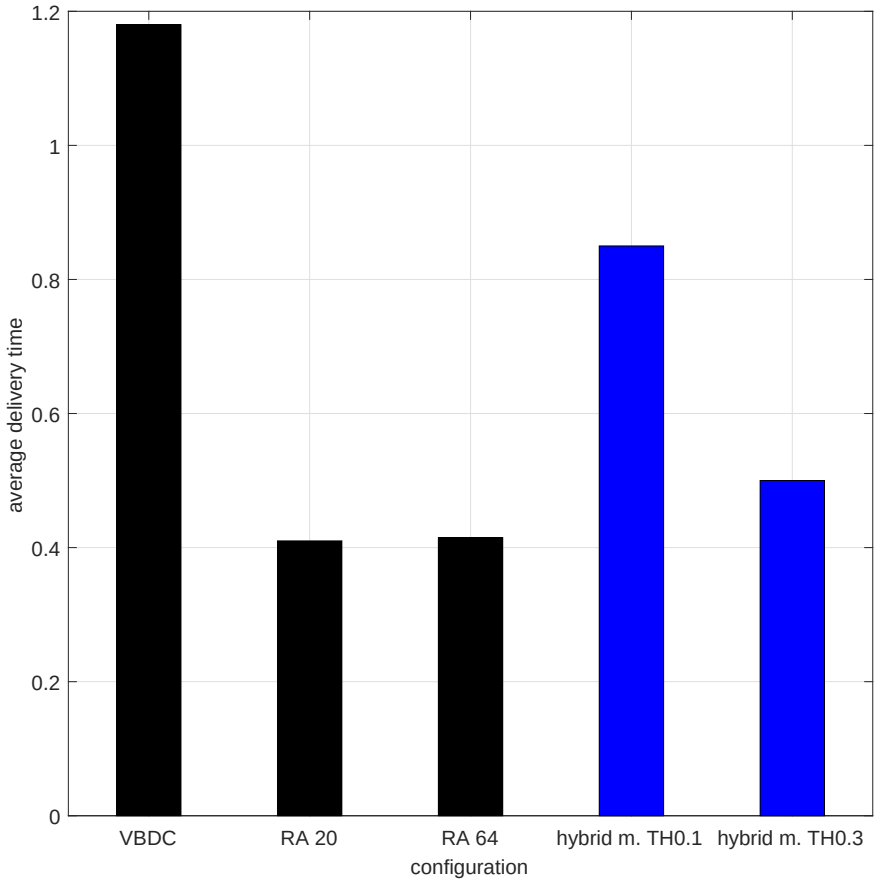


Figure 4.11: Average delivery time [s] of the different configurations under consideration.

RA block, which has no access delay.

Finally, the average number of requested DA time-slots per superframe per configuration is shown in Figure 4.12: we recall that *RA 20* and *RA 64* configurations do not rely on any DA requests, thus they are not shown here. The VBDC value of about 3.3 time-slots per superframe can be considered as a reference, since it achieves 100% of resources utilization. In the *hybrid mode*

Table 4.9: Sensitivity to *BLRthr* parameter: actual simulated BLR and average delivery time

BLRthr	**BLR**	**Average Delivery Time [s]**
0.1	0.34	0.89
0.15	0.4	0.82
0.2	0.5	0.68
0.25	0.54	0.56
0.3	0.56	0.51

configurations, the number of required DA time-slots decreases, because only traffic spikes are transmitted relying on DA time-slots. Figure 4.12 shows that, for an increasing BLR_{thr}, less and less DA time-slots are requested, at the price of an increasing BLR. Note that a fraction of DA time-slots is unused, thus opening to the possibility to add additional non-DA sources if necessary.

4.2.5 Considerations and discussion

The empirical assessment of the achievable performance level, when using the hybrid access scheme proposed in this work, shows how the discussed parameters can be properly tuned to satisfy different sensor application requirements, in terms of desired delivery time and tolerable BLR. The performance evaluation shows the sensitivity of the algorithm to the desired BLR threshold, BLR_{thr}: if the threshold is overcome, one or more RCSTs are forced to switch to the use of DA. The latter decision can be taken locally by each RCST, if the buffer size has grown larger than the local threshold Q_{RA}, or centrally by NCC, if $BLR > BLR_{thr}$: both cases reduce the contention on the RA block, thus lowering BLR. A RCST that has been forced by NCC to use only DA time-slots can re-enable RA operation mode based on a local decision: if the queue length is less than a predefined threshold, namely Q_{RA}, RA operation mode can be re-enabled. In this way, future short data bursts can be delivered via RA operation mode, which shows a lower delivery delay, at the price of

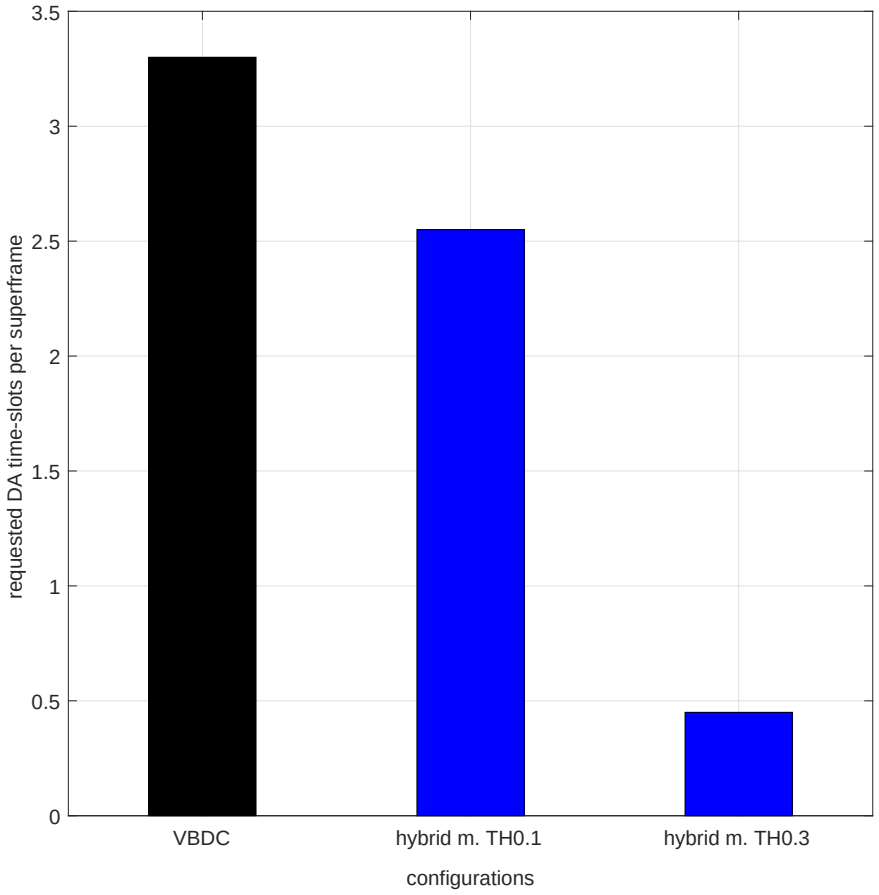


Figure 4.12: Average number of requested DA time-slots per RCST per superframe

possible collisions. In fact, the average delivery time depends on BLR_{thr}: a larger BLR_{thr} implies a lower delivery time, and vice-versa.

The proposed hybrid access scheme allows the coexistence of RCSTs showing different traffic patterns: from eager ones, as a greedy RCST can exhibit, to sporadic IoT/M2M traffic profiles. The access scheme proposed in this work privileges the latter one in terms of delivery time, while forcing the former ones to rely on a DA assignment: both traffic patterns benefit of this strategy,

because short data bursts are immediately delivered, while longer ones can complete the data transfer in the shortest possible time thanks to the dedicated time-slots. The available parameters allow a fine tuning, which in turns balance the trade-off between the delivery time and the desidered BLR_{thr} value.

Chapter 5

Reliable delivery of IoT / M2M data via RA Satellite Links

This Chapter describes the second contribution of this Ph.D. thesis: it discusses the cross-layer interactions between IoT/M2M TCP-based protocols and an underlining CRDSA++ access scheme. The focus of the scientific community has been so far mostly around the implications of RA on physical layer design, only partly addressing its interactions with upper layers (e.g., TCP/IP stack). In particular, the performance analysis of TCP-based higher layer protocols/services running on top of RA schemes has not yet provided meaningful indications about the boundaries of its applicability. Furthermore, the impact of collision resolution on TCP congestion control in terms of delay as well as the reaction of TCP in response to TCP segments' collision still needs a rigorous investigation from a theoretical standpoint. The following work attempts to bridge this scientific gap, by analysing the performance of TCP-based application protocols over CRDSA++, running in a DVB-RCS2-like network under different traffic load regimes. MQTT[1] is used in the following as reference protocol because of its large diffusion; furthermore, because of the *broker* node, an entity of the PUB/SUB architecture, this choice is

[1]MQTT is described in Section 2.3.1.2.

advantageous in satellite networks because, if placed just before the satellite, it splits the TCP connection, similarly to a Performance Enhancement Proxy (PEP). The use of UDP-based protocols, as for instance the CoAP protocol[2], is not taken into account, because of the absence of a reliable communication: in fact, the reliability of the CoAP protocol is only optionally available at the application layer and implements a simple Stop-and-Wait ARQ.

Section 5.1 shows the behaviour of MQTT on top of CRDSA++, when very short and short IoT/M2M connections are considered. Empirical results are available, and the performance metric under consideration is the *completion time*, here defined as the time it takes a source to successfully transfer a variable amount of data via TCP; furthermore, simple but efficient ways to reduce the completion time are discussed and empirically proved. The scenario in Section 5.2 considers a different scenario: several terrestrial Wireless Sensor Networks (WSNs) generate large amounts of IoT/M2M data to be sent via a RA satellite link, thus taking into account a sustained traffic profile, instead of a sporadic one. Thus, the metric under consideration is the throughput experienced at the application layer; MQTT relies here on TCP NewReno at transport layer. To correctly estimate the throughput at the application layer, NewReno throughput must be estimated, because the congestion control algorithm can limit the achievable throughput at the application layer. Because of the error in the throughput estimation when relying on the theoretical models in literature [51, 52, 53], a refined TCP NewReno throughput estimation model in RA satellite channels is here proposed and validated; further to this, the impact of segment losses due to collisions on the dynamics of the congestion control algorithm is analysed, and a simple but effective model is proposed and validated, aiming at binding the effects of layer-2 losses w.r.t. layer-4 events. Finally, the stability guaranteed by the use of TCP in RA channels is discussed and empirically proved, removing the need for load control algorithms at MAC layer, which can prove difficult to be properly tuned in different scenarios.

[2]CoAP is described in Section 2.3.1.1.

5.1 Advances on Elastic Traffic via M2M Satellite User Terminals

Owing to the variety of traffic profiles that M2M and IoT applications may generate, this work studies the applicability of CRDSA when clusters of sensor nodes exchange data via satellite toward a sink via short-lived TCP/IP connections. In the scenario under consideration, uncorrelated sensor sources generate IoT/M2M data, which are multiplexed by a RCST and sent as MQTT payload. We aim at investigating the scalability in terms of number of RCSTs that can successfully transfer data with small delivery delays, in order to avoid congestion phenomena at MAC layer, which may induce collision storms.

5.1.1 Introduction

It has been over a decade since people started to talk about IoT. Today it is a reality, with many of us surrounded by several Internet-capable devices wherever we are. A lot of work has been recently done in standardization groups (e.g., IETF) on various aspects surrounding the IoT, for instance what has been done by the CoRE working group. The outcomes of these efforts are summarized in Section 2.3.

The PUB/SUB paradigm, exploited in several architectures, is based on a messaging pattern where the senders, namely *publishers*, do not send the messages directly to specific receivers, namely *subscribers*, but rely on a intermediate entity, usually referred to as *broker*. MQTT is a IoT/M2M protocol designed to be extremely lightweight; it is useful in presence of connections requiring a small code footprint and/or when network bandwidth is at a premium; the broker is responsible for collecting the subscribers' interests on available topics and for delivering new data coming from connected publishers. For instance, MQTT has been selected as the most promising protocol in healthcare scenarios where a large number of devices communicate with a mobile broker and in a wide range of home automation and small device scenarios [54]. Since MQTT relies on TCP as transport protocol, the achievable performance in RA satellite links should be carefully evaluated, because TCP performance can be poor in improperly configured fat links. In the case of DVB-RCS2, which is versatile to a large variety of applications in broadband

satellite communications, the coupling of TCP short-lived connections and RA schemes requires further investigations. In fact, nowadays, short transfers dominate most of Internet traffic: this is true, particularly, when referring to the quickly arising IoT/M2M traffic type. A fraction of this traffic shows a large sensitivity to delays because of its quasi-interactive nature. To ensure rapid transfers, TCP should work in the Congestion Avoidance (CA) phase [55], or *steady state*; anyway, short transfers will unlikely enter CA because the data transfer may be completed before that or because the collisions may force TCP in the Slow Start (SS) phase. Further to this, the aggressiveness of TCP during the SS phase may lead to an undesired striking collision rate, thus severely impairing the performance of TCP connections.

The main contribution of this part is the characterization of the *completion time* of a MQTT data transfer: in other words, evaluating how much time a publisher needs to successfully complete a data exchange with the broker. Indeed, the main interests are in:

- assessing the performance of MQTT on CRDSA++ links;
- evaluating the possibility to reduce the completion time by assessing the impact of the initial SSTHR value and the transmission buffer size.

The rest of this Chapter is organized as follow: Section 5.1.2 gives insights on the state of the art of the literature in this context; Section 5.1.3 describes the scenario under consideration and Section 5.1.4 shows and discusses the simulation results. Finally, the conclusions are drawn in Section 5.1.5.

5.1.2 Related Works

Since the Internet traffic is highly dominated by short transfers [56], a great effort has been directed to TCP short flows - also called *mice* TCP - modelling. Cardwell et al. extended the analytical model for such an elastic traffic [57], which was firstly proposed in [58], in order to characterize the startup phase of TCP connections. Mellia et al. proposed an analytical model to predict the TCP performance in terms of completion time for short-lived flows [59]. Based on the knowledge of the segment loss probability, the average RTT and the flow length (in number of segments), the analytical model in

[59] provides a good approximation of the average completion time. A critical factor to reduce the completion time of short data transfer is the initial SSTHR, which should be determined to achieve a reasonable trade-off. In fact, a too low setting of the initial SSTHR value forces TCP to prematurely switch in CA, forcing the sender to probe for available network bandwidth instead of overfeeding it; on the other hand, a too high setting of the initial SSTHR value forces the Congestion Window (CWND) to quickly grow beyond the link capacity, thus causing multiple packet loss events that trigger time-consuming Retransmission Timeouts (RTOs). An optimal setting of the initial SSTHR value should be close to the Bandwidth-Delay Product (BDP), which maximizes the overall performance of all TCP connections sharing the bottleneck, and it is defined as the ratio of the nominal transmission delay and the spacing of packets at the bottleneck:

$$BDP = \frac{delay * bandwidth}{MSS}. \tag{5.1}$$

As preliminarily shown in [60], mice-TCP connections generated by IoT/M2M traffic, as in the case of MQTT, may significantly be impaired by long latency links and contention access medium. In fact, TCP may be able to self-regulate its transmission rate on contention access channels, facing collisions on the contention medium as well as congestion loss events [21]. In the following, the impact of the satellite latency and fat lossy links in presence of short-lived TCP flows is analysed. Such a traffic is sent on the return link of a DVB-RCS2 system, towards the broker node, here acting as a *sink*.

5.1.3 Proposed scenario

Consider a system with a finite set of nodes (i.e. RCSTs) over a RA channel. Time is slotted and each superframe contains one RA block, spanning the whole superframe. Each frame is composed of n time-slots and each RCST can exploit any transmission opportunity for sending data. The DVB-RCS2 standard offers several WaveForms (WFs) to deal with different channel conditions and traffic types: the redundancy added by channel coding schemes can be tuned according to channel statistics; the slot size can also be different to adapt to different traffic types, such as small payloads for M2M-like traffic, or large payloads. Physical layer parameters are here chosen accordingly to

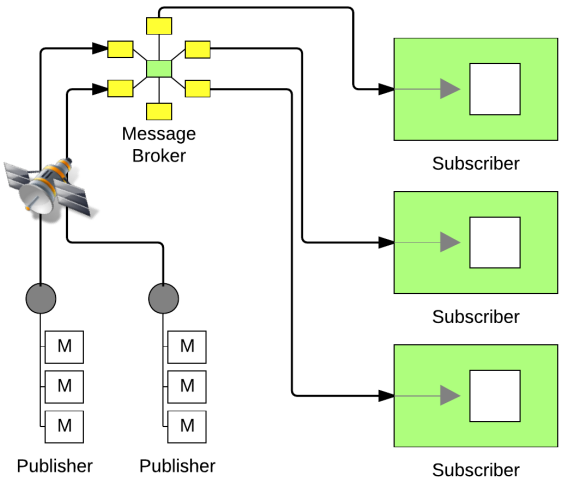


Figure 5.1: Logical description of the proposed scenario: publishers sending MQTT data to a remote broker via satellite

those defined for WF 14[3], leading to a gross slot size of 188 [B]; this WF provides the largest layer-II burst size, which minimizes the fragmentation of IP datagrams that encapsulate TCP segments.

In the scenario under consideration, a segment loss event can be triggered only by collisions and by buffer overflows. The return/forward links are supposed ideal and each ACK is supposed correctly received. No ARQ algorithm is in use at MAC layer. Each TCP segment sent to lower layers, i.e., network layer first and MAC layer then, is queued in a finite buffer, here referred to as *MAC queue* of size 32 [IP datagrams]. Table 5.1 recalls numerical values here in use.

The traffic arrival process for each RCST is driven by a MQTT publisher, sending messages to the broker, as shown in Figure 5.1. A single application runs per RCST; TCP NewReno is used at transport layer and *delayed ACKs* are not in use ($b = 1$) . The Maximum Segment Size (MSS) is 486 [B], since this size fits exactly into an integer number r of slots, being $r = 3$. The data transfers may be short enough to suffer a few packet losses or zero losses. The

[3] *Table A-1: Reference WFs for Linear Modulation Bursts*, Digital Video Broadcasting (DVB); Part 2: Lower Layers for Satellite standard, DVB Document A155-2, January 2013.

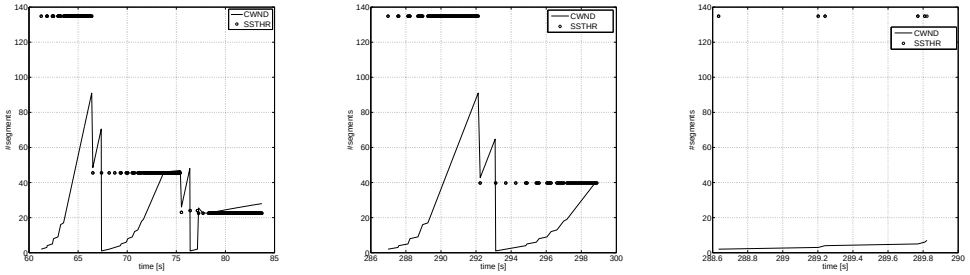
Name	Value
TCP flavor	Reno / NewReno
TCP MSS	486 [B]
TCP/IP headers size	40 [B]
TCP header options size	12 [B]
MAC queue size	32 [IP datagrams]
RA scheme	3-CRDSA
RA blocks per superframe	1
RA block duration	0.013 [s]
Slots per RA block	64
Gross slot size	188 [B]
Net slot size	182 [B]
Bandwidth	8012820 [Hz]
Roll off	0.2
Carrier spacing	0.3 [Hz]
One-way delay	0.13 [ms]

Table 5.1: System parameters of the simulator

process of publishing new data begins with the application requesting the opening of a TCP connection, which starts the 3WHS procedure. If the latter fails, several tries are made and the timeout interval increases; most TCP implementations abort the opening after 4-5 failures. Here, the connection is supposed always successfully opened, usually after the first *active open*[4] request. Once the connection is successfully set up, the data transmission starts; when the data transfer ends, the connection is *actively closed*[5]. In these scenarios, each application enqueues data in the TCP sending buffer rapidly enough to allow for TCP to send data as fast as the CWND allows. The traffic profile under consideration has the data payload length randomly

[4]An *active open* request is made by a client that sends a SYN message to start the connection; a *passive open*, instead, means that the server starts a listening process and waits for connections.

[5]A client which sends a FIN notification to the server is *actively closing* the connection; instead, the side receiving the FIN notification is said to *passively close* the connection.



(a) Long connection x_{f1} [B] transferred

(b) Medium connection x_{f2} [B] transferred

(c) Short connection x_{f3} [B] transferred

Figure 5.2: CWND and SSTHR of TCP connections of different lengths as opened by an MQTT-based application

drawn according to three different Pareto distributions of fixed shape $\alpha = 1.1$ and average values $x_{f1} = 524888$ [B], $x_{f2} = 104858$ [B] and $x_{f3} = 10240$ [B]. The three different Pareto distributions are here meant to simulate different application payload lengths, in order to replicate an aggregate M2M traffic load as close as possible to real scenarios, as discussed in Section 2.2. One third of the overall MQTT sessions has payload length randomly drawn from x_{f1}, one third from x_{f2} and the last third from x_{f3}. Figures 5.2a, 5.2b and 5.2c shows the CWND and the SSTHR values for these three different classes of connections, showing how very short connections (in Figure 5.2c) quickly complete the data transmission and without experiencing RTOs or buffer overflows, on average; on the other hand, as the payload size increases (in Figures 5.2a and 5.2b), the collisions on the RA channel trigger the Fast Recovery/Fast Retransmit (FR) phase and/or RTOs to recover lost segments, thus forcing CWND to smaller values, on average.

5.1.4 Performance evaluation

The performance evaluation shown in the following is based on the NS-3 [7] simulator, extended so as to implement a DVB-S2/RCS2 satellite network, according to the modules described in [6]. The aim is in characterizing the completion time of IoT/M2M connections over RA satellite channels. In Section 5.1.4.1, the simulation results are compared with the analytical model

in [59], considering payload lengths up to 4.3 [kB]; then, in Section 5.1.4.2, simulation results are shown for payload lengths up to 120 [kB].

5.1.4.1 MQTT connections with payload length up to 4.3 [kB]

The analytical model in [59] allows to estimate the completion time of very short TCP connections: C_m^1 is the completion time for a connection starting with CWND equal to 1 MSS (default value) and enqueuing for transmission m segments. The model in [59] is limited to $m \leq 9$ because of the too large complexity of the model when m increases; the estimation accuracy decreases as m increases. The model is based on a recursive approach, that can be summarized as: $C_m^1 = C_1^1 + C_{m-1}^2$, where the completion time for a connection starting with a CWND equal to 1 MSS and transmitting m segments is equal to the time to transmit a single segment, when the CWND is equal to 1 MSS, plus the time to transmit the remaining $(m-1)$ segments with a starting CWND equal to 2 MSS. For instance, $C_3^1 = C_1^1 + C_2^2$ and, if no losses are experienced, two RTTs are needed to complete the transmission plus one RTT to complete the 3WHS (the first segment is sent out together with the ACK actually starting the data exchange). The model in [59] also accounts for 3WHS time, C_s, but does not model the *active close* phase. The completion time of very short NewReno-based MQTT connections ($m \leq 9$) has been validated against the aforementioned model, to verify its applicability in this context: the results are shown in Figure 5.3, which shows the completion time versus the number of transmitted segments. The model in [59] provides an overall good match between the simulation results and the analytical model; as anticipated, the error is larger if m approaches the value of 9 MSS.

5.1.4.2 MQTT connections with payload length up to 120 [kB]

To the best of the author's knowledge, there is no model in literature able to estimate the completion time of TCP connections transferring more than 9 segments. In this Section, an empirical evaluation is provided, which also compares the performance of Reno and NewReno, in order to prove a shortcoming of the NewReno FR implementation in presence of short data transfers. Figure 5.4 shows that the completion time of NewReno-based MQTT

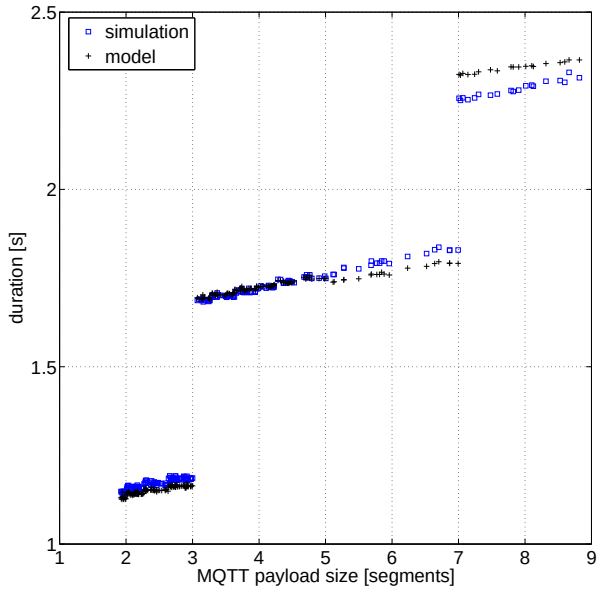


Figure 5.3: Completion time (excluding *active open* and *active close* phases) for very short TCP connections, as in Mellia's model

connections is almost linear w.r.t. the length of the transmission. But a sudden inflation in the completion time is visible in a specific range of payload lengths: for instance, if considering a MAC queue size equal to BDP, the completion time change its slope for a payload size comprised between 50 and 100 segments; the same goes when using a MAC queue with size twice BDP and a payload length comprised between 80 and 130 segments. This phenomenon is due to *tail losses*[6], and the FR mechanism is responsible for the inflation of the completion time. A tail loss can be explained as follows: if a burst of M losses occurs at the tail of the transmission[7], the connection enters the FR phase, whose duration is equal to M [RTT], if no new losses occur: in fact, the NewReno FR implementation allows to recover a single segment per RTT.

[6]Tail losses are described in the IETF draft available at: `tools.ietf.org/html/draft-dukkipati-tcpm-tcp-loss-probe-01`.

[7]In the scenario proposed in this Chapter, a burst of losses can occur because of the MAC queue size, which may be responsible of buffer overflows.

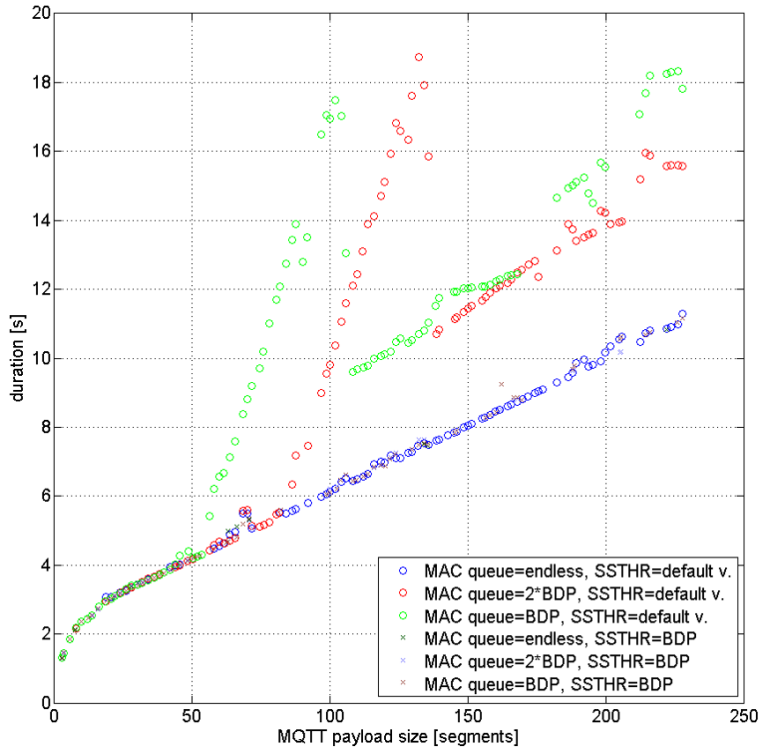
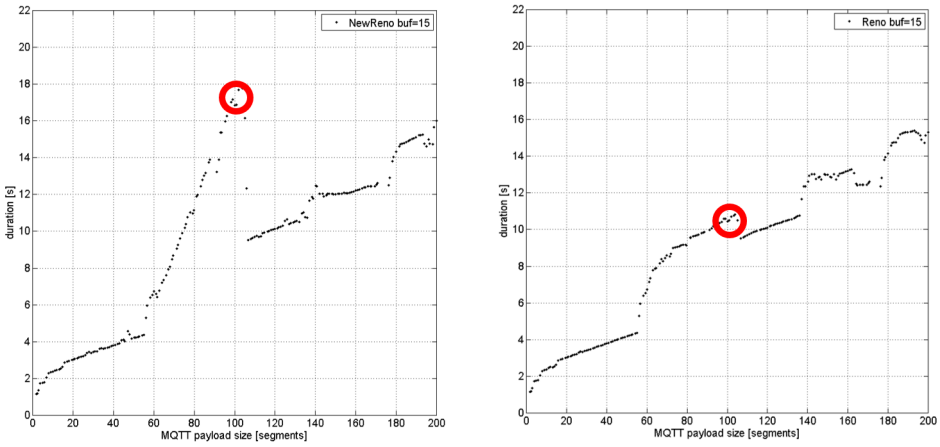


Figure 5.4: Completion time of NewReno-based MQTT connections: depending on the MAC queue length, a sudden inflation of the completion time is visible for payload lengths between 50 and 100 segments.

In the following, the completion time to deliver 100 segments ($\approx$ 47.5 [kB], marked with a red circle in Figures 5.5, 5.6, 5.7 and 5.8) is used as a comparison metric to describe the effect of tail losses; furthermore, possible strategies to avoid it are discussed. Figure 5.5 shows the completion time of NewReno and Reno connections, along with the tail loss effect: in Figure 5.5a, the tail loss is clearly visible, and the completion time for 100 segments is $\approx$ 17 [s], while Figure 5.5b shows that the completion time for 100 segments is $\approx$ 11 [s], when using Reno. Thus the use of NewReno in presence of really short bursts can lead to undesired effects and, even if it may result counter-intuitive, forcing a RTO can be less time-consuming than recovering a single lost segment per RTT.

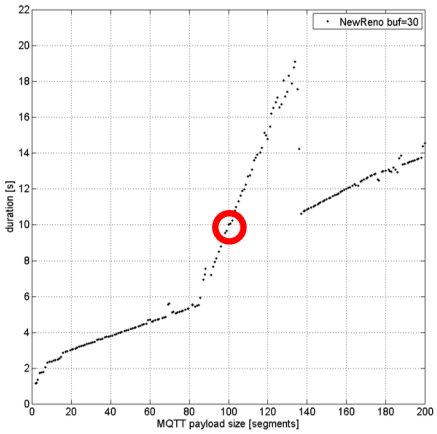


(a) TCP NewReno (b) TCP Reno

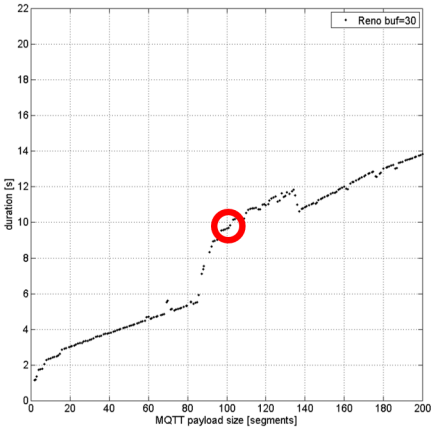
Figure 5.5: Completion time for NewReno and Reno connections: 100 segments delivered in ≈ 17 [s] (NewReno) / ≈ 11 [s] (Reno)
-MAC queue size equal to BDP and initial SSHTR at default value-

If the MAC queue size is doubled, the segments range subject to the aforementioned inflation is shifted, as shown in Figure 5.6, and this leads to a completion time of ≈ 10 [s] for 100 segments: note that this is a fortuitous case, because just varying the transmission length would lead to different completion times for Reno and NewReno flavours. Anyway, tail losses, when due to buffer overflows, can be avoided by relying on very large buffers, as demonstrated in Figure 5.7, where is visible that Reno and NewReno show the same exact behaviour and 100 segments are delivered in ≈ 6.2 [s]. Eventually, the effect of a different setting of the initial SSTHR value is shown in Figure 5.8: setting the initial SSTHR to BDP is sufficient to avoid the tail losses effect, independently of the MAC queue size. Figures 5.8a and 5.8b show, respectively, that the performance level of NewReno and Reno is the same and, furthermore, that the completion time is the lowest one: ≈ 6 [s].

Finally, in Table 5.2, the experienced RTTs by NewReno and Reno connections are compared under the aforementioned conditions. In the first column, the size of the MAC queue and the initial SSTHR value can be read, where each row defines a specific sub-scenario: the first row shows the typical ope-

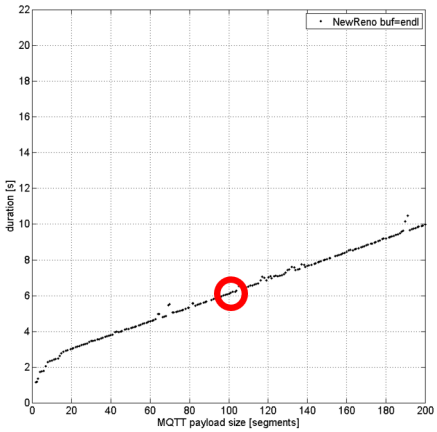


(a) TCP NewReno

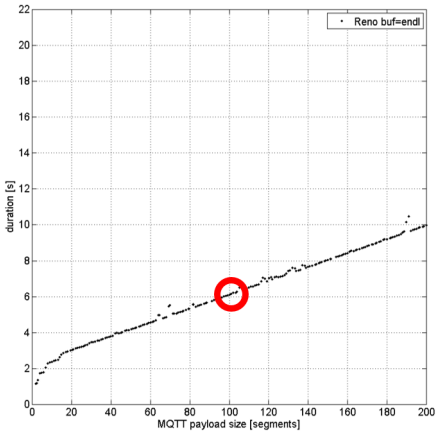


(b) TCP Reno

Figure 5.6: Completion time for NewReno and Reno connections: 100 segments delivered in ≈ 10 [s] (NewReno) / ≈ 10 [s] (Reno)
-MAC queue size equal to 2BDP and initial SSHTR at default value-

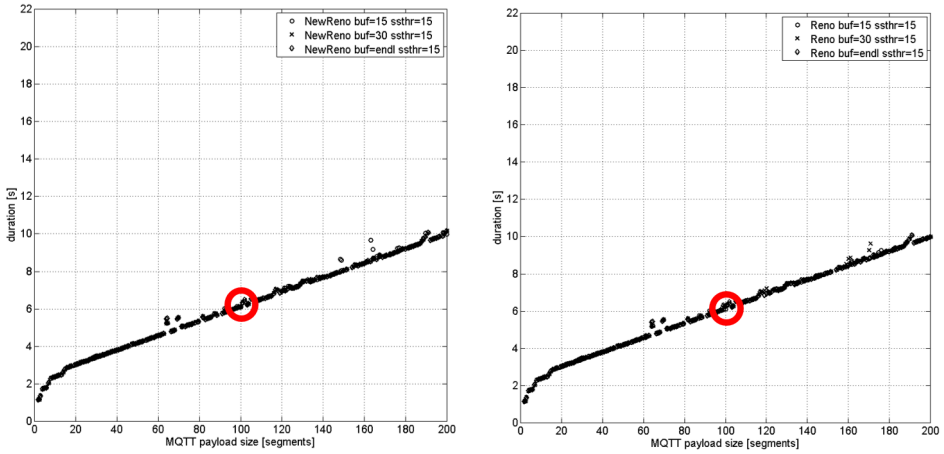


(a) TCP NewReno



(b) TCP Reno

Figure 5.7: Completion time for NewReno and Reno connections: 100 segments delivered in ≈ 6.2 [s] (NewReno) / ≈ 6.2 [s] (Reno)
-MAC queue virtually endless and initial SSHTR at default value-



(a) TCP NewReno (b) TCP Reno

Figure 5.8: Completion time for NewReno and Reno connections: 100 segments delivered in ≈ 6 [s] (NewReno) / ≈ 6 [s] (Reno)
-increasing MAC queue sizes and initial SSHTR at BDP value-

rating settings, a large buffer (here considered virtually endless because of the short data bursts) and the default value of the initial SSTHR = 64 [kB] ($\approx$ 135 segments of size 486 [B]). The mean value, and the 0.25 and 0.75 quantiles of the RTT are shown per TCP flavour: the best results are in the last row, when both MAC queue size and initial SSTHR values are equal to BDP. Anyway, this configuration offers a small advantage w.r.t. the configuration in the third row, when only the MAC queue size is equal to BDP; in fact, a proper configuration of the buffer size is enough to ensure almost optimal performances, because the CWND size would oscillate between BDP and 2BDP, i.e., the optimal operating region [55]. But both these solutions can be tricky to realize because the buffer size should be varied, and the hardware components of a net adapter cannot be modified every time a node works under different conditions; instead, using buffers as they are (large, nowadays) and modifying a software setting, i.e., the initial SSTHR value, as in the forth row, leads to a performance level close to the optimal one. Thus, when considering short data bursts, typical of IoT/M2M traffic, an optimal strategy to effectively reduce the completion time is to just set the initial SSTHR value

Table 5.2: NewReno vs Reno RTTs as responsiveness metric.

MAC q. size, in. SSTHR v.	**NewReno RTT** [s]			**Reno RTT** [s]		
	aver.	0.25q	0.75q	aver.	0.25q	0.75q
endless, default	3	1	5.2	3	1	5.2
2BDP, default	1.2	0.8	1.66	1.2	0.76	1.6
BDP, default	0.88	0.68	1.1	0.78	0.6	0.97
endless, BDP	1	0.7	1.2	1	0.7	1.2
2BDP, BDP	0.9	0.7	1.1	0.9	0.7	1.1
BDP, BDP	0.77	0.65	0.9	0.77	0.66	0.9

equal to BDP. A large buffer should be used, in order to avoid any overflow effects; but large buffers are common nowadays, so complying with the latter guideline does not pone any inconvenience.

5.1.5 Considerations and discussion

In this Chapter, the performance level of MQTT via RA satellite channels has been assessed: short and sporadic data transfers are taken into account, thus the chosen metric is the *completion time*. An empirical assessment of the completion time has been provided for short, medium and large IoT/M2M data transfers, up to 120 [kB]. The undesired effects of the NewReno FR implementation have been highlighted, which cause tail losses responsible of a longer completion time in presence of short data bursts. To prove it, this work shows the completion time when using the Reno FR implementation: it shows how triggering a RTO can be more effective than recovering a single segment per RTT, in presence of short data bursts. Furthermore, the more convenient technique to reduce the completion time has been shown, i.e., properly setting the initial SSTHR value, which is more feasible, in real scenarios, than adjusting the transmission buffer size. Note that the use of a larger initial CWND is not taken into account here, and it is left out as future work, as well as the modelling of the completion time for TCP transfers larger than 9 segments.

In Section 5.2, a different key metric is taken into account to evaluate the

performance level of MQTT in the same operative conditions: the throughput, as experienced at the application layer. In fact, to avoid the *tail losses* inconvenience while still using NewReno, aggregation techniques can be used at the broker before delivering data to the subscribers, in order to consider the alternative scenario of a sustained offered load.

5.2 M2M Traffic via Random Access Satellite links: Interactions between Transport and MAC Layers

M2M services are witnessing an unprecedented diffusion, which is expected to result in an ever-increasing data traffic load. In this context, satellite technology is playing a pivotal role, since it enables a widespread provisioning of M2M services. In particular, oil industry, maritime communications, as well as remote monitoring are sectors where the use of satellite communications is expected to dramatically explode within the next few years. In the light of this sudden increase of M2M data transported over satellite, a more thorough understanding of M2M service implementation over satellite is required, especially focusing on the interaction between transport and MAC layers of the protocol stack. Starting from these observations, this Chapter thoroughly analyses the interaction between TCP and CRDSA, assuming the use of an MQTT-like protocol to distribute IoT/M2M services. A novel TCP model is developed and validated through extensive simulation campaigns, which also shed important lights on the design choices enabling the efficient transport of M2M data via satellite.

5.2.1 Introduction

According to data traffic forecast reports, the volume of data transported by Internet in 2019 will exceed the threshold of 2.0 zettabytes per year, generated by more than trillion of devices. Only a minor portion of the traffic will be generated by PCs, as commonly observed in the recent past; on the contrary, a large quota of Internet traffic will be generated by TVs, tablets, smartphones, and M2M devices. In particular, it has been highlighted that M2M traffic will experience a growth rate in the order of 60%. The large amount of traffic contributed by M2M applications will have an important impact on the design of future network architectures and on dimensioning the capacity of the telecommunication infrastructures. In more words, M2M applications are today largely diffused in several terrestrial deployments and have pushed the scientific community to thoroughly investigate network de-

sign implications. The survey in [3] underlines that current M2M markets are fragmented, because various vertical M2M solutions have been already designed and implemented: the efforts to identify a common architecture is of a primary concern. Moreover, reference [3] also shows that typical IoT/M2M services exhibit intermittent behaviour (e.g., ON-OFF oscillations), low data-rates, and high traffic burstiness, thus motivating the use of RA schemes. Seminal works about the use of RA for the efficient transport of M2M services via satellite are provided in [12, 13]. The use of satellites is expected to become crucial in the near future [1, 2], because very highly dense networks, such as M2M ones, pose several challenges to terrestrial wireless networks [61]. In fact, the availability of satellite connectivity is of paramount importance in scenarios such as remote plant monitoring, ship tracking service, and aeronautical telemetry transmission, just to mention a few, where a typical terrestrial infrastructure cannot be exploited. From this standpoint, a special note has to be reserved to the case of M2M services distributed via satellite, whose related industry is continuously increasing in size.

The design of efficient RA schemes for satellite communications has captured a great interest from the scientific community and the satellite industry over the last decade [12, 13]. An important example is represented byDVB-RCS2, which specifies the transmission schemes and the protocol architecture for the return link, where CRDSA is proposed as a complementary method to the traditional DAMA. CRDSA offers immediate access to satellite capacity without incurring in the reservation delays of DAMA that can penalise the transmission of bursty traffic (e.g., M2M).

The focus of the scientific community has been so far mostly around the implications of RA on physical layer design, only partly addressing its interactions with upper layers (e.g., TCP/IP stack). In particular, the performance analysis of TCP-based higher layer protocols running on top of RA schemes has not yet provided meaningful indications about the boundaries of its applicability, apart from the contribution in [60], which anyway offers a simple and preliminary empirical evaluation. Furthermore, the impact of the collision resolution scheme on TCP congestion control in terms of delay as well as the reaction of TCP in response to TCP segments' collision still needs a rigorous investigation from a theoretical standpoint. This paper attempts to bridge

this scientific gap, by analysing the performance of TCP-based application protocols over CRDSA++, using 3 replicas and running in a DVB-S2/RCS2 network under different traffic load conditions. To this end, a rigorous analytical model of TCP dynamics over RA is drawn, adapting and extending the TCP models in [51, 52, 53] that cannot inherently apply to the case of RA schemes. The analytical approach will be validated via NS-3-based simulations [7].

The remainder of this Chapter is structured as follows: Section 5.2.2 surveys the main findings on TCP modeling and interactions with RA schemes, paying special attention to the case of satellite networks. Section 5.2.3 introduces the reference system, while Section 5.2.4 proposes some refinements that are necessary on long-delay RA links in order to correctly estimate TCP throughput. Section 5.2.5 offers a simple but effective model to link together burst losses at MAC layer, due to collisions, and segment losses at transport layer. The performance analysis is carried out in Section 5.2.6 and, finally, the conclusions are drawn in Section 5.2.7.

5.2.2 Overview

5.2.2.1 Related works

In this Section, a survey is presented about the theoretical models available in the literature for estimating the TCP throughput. The literature about TCP modelling can be divided into two main sets, characterized by different packet loss models: bursty losses [53, 57] and independent losses[8] [62, 63]. The TCP behaviour is modelled as a *renewal process* for the first time in [53] and then in [63]. A TCP *cycle* is defined as the period of time between to consecutive loss events and it is equivalent to an *epoch* of the corresponding renewal process. The performance of TCP is modelled in terms of *rounds*: an epoch contains m rounds. A round begins with the transmission of W segments, called CWND, and ends upon the receipt of the ACKs corresponding to these segments. The duration of a round is assumed to be independent of the CWND size and dependent on the RTT. The CWND grows at each round,

[8] As far as contention-based medium access schemes (e.g., CRDSA) are concerned, losses due to collisions should be considered as independent events: this is empirically proved in Section 5.2.6.

until a loss event occurs; then, TCP enters the recovery phase. Apart from the recovery algorithm that may vary according to the TCP flavor, TCP leaves the recovery phase with or without the expiration of the RTO, thus entering a new epoch. The first rounds of the new epoch correspond to either a SS or a CA phase, the former if RTO has expired, the latter otherwise. In [53], a Triple-Duplicate ACK (TD) is used as loss indication and a TD Period (TDP) (i.e., the period between two TD loss indications) coincides with an epoch of the renewal process. Since our aim is to establish a relationship between the throughput T of a TCP connection and the segment loss probability p, let S_i be the number of TCP segments sent during the i-th epoch and D_i the duration of that period. Thus, we can express the TCP throughput as follows:

$$T = \frac{\mathbb{E}[S_i]}{\mathbb{E}[D_i]}. \tag{5.2}$$

The main outcomes in [53], then refined in [64], are the so-called Square Root Formula (SRF) and the Approximated Model (AM). The model in [53] treats segment losses as independent events in different rounds and as correlated events in the same round: each segment delivered after the first segment loss is supposed to be lost as well, independently of its actual outcome. The throughput models in [53] have been derived for TCP Reno recovery algorithm, which can recover a single loss during FR. However, more recent TCP implementations, like NewReno or Cubic, have more complex error recovery mechanisms that allow recovering multiple losses during the FR phase [65]. Other works have extended SRF and AM in order to account for FR phase' dynamics with Slow But Steady (SBS) flavor [51, 52]. The most prominent innovation in [51] and [52], with respect to [53], is that both bursty and independent losses are modelled.

When specifically targeting the use of TCP in RA schemes for satellites, the work in [66] provides upper and lower bounds of the throughput for a simplified TCP model in a SA channel. Furthermore, TCP performance over a satellite DSA medium is analysed for a single TCP connection in [67]; the authors assume that the MAC layer operates at a target load G^* and that the NCC periodically broadcast an *activity factor* (i.e., a throttling parameter) to dynamically adjust the proposed flow control policy applied at RCSTs, in order to maximize TCP throughput. Analogously, [68] analyses TCP

throughput over CRDSA when packet-level Forward Error Correction (FEC) techniques are employed; instead, in this work, we do not use any packet-level FEC techniques. Furthermore, a single TCP connection is modelled through a fluidic model in [68], less precise than the packet level simulator in use in this work. Finally, reference [18] considers a similar scenario, i.e. the use of MQTT in RA satellite channels, but it focuses on a different metric, that is the *completion time*. In fact, the broker is placed close to the subscribers, thus each publisher delivers its data via satellite to the remote broker. Furthermore, a simplified scenario is taken into account: low/medium traffic conditions are under consideration and the impact of packet fragmentation is not analysed.

5.2.2.2 Main contributions

This work extends the contribution in [24], where the problem was firstly stated and discussed. Here, a complete set of MAC parameters is taken into account in the analysis to better describe the behaviour of TCP over a CRDSA-based satellite channel. This work also proposes to use TCP for sending IoT/M2M traffic over CRDSA++, removing the need for any other flow control scheme, ARQ and load control algorithms in use at MAC layer.

To summarize, the main contributions in this work are:

- the analysis of how the TCP congestion control algorithm behaves in a channel dominated by collisions, where TCP auto-regulates the sending rate to counteract an increase in Segment Loss Rate (SLR);
- a finer model derived from [51, 52, 53], which accurately matches the TCP steady state throughput over RA satellite channels;
- a simple but effective model establishing a relation between the loss rate at MAC layer and the loss rate at transport layer (cross-layer study);
- finally, considerations on system stability guaranteed by the TCP control congestion algorithm, even at very high loads.

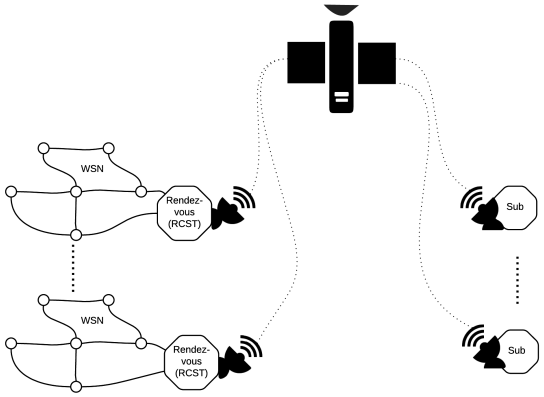


Figure 5.9: Scenario under investigation: multiple RCSTs acting as rendezvous nodes and delivering data to remote subscribers via RA satellite link

5.2.3 System Architecture and Setup Parameters

5.2.3.1 Reference scenario

This work focuses on the applicability of the DVB-RCS2 standard to scenarios in which IoT/M2M traffic is generated from WSNs, similarly to what is considered in [5]. The collected data are *published*, according to the PUB/SUB paradigm; a *broker* node collects data and makes them available to *subscribers*. In the scenario depicted in Figure 5.9, each WSN has a local broker node (also referred to as *cluster head* or *rendezvous* node); the subscribers are connected via a satellite link. Several WSNs are deployed on the ground and each WSN has a potentially large number of sensors, which periodically generate IoT/M2M data. Each broker has no constraints about power consumption and available computational resources; it is connected to a RCST, sending collected data via an RA satellite channel.

The broker is responsible for the delivery of data collected by the WSN. The broker is supposed to have an endless queue of IoT/M2M data waiting for transmission, so that the offered traffic is sustained. Such assumption is valid in several possible application scenarios: for instance, in battle-field communications, where several disjoint teams can use several WSNs sending and receiving data from a remote command centre; in maritime scenarios

for fleet management; in Flying Ad-Hoc Networks (FANETs), where several resource-constrained Unmanned Aerial Vehicles (UAVs), composing a swarm, can rely on a local rendezvous node or acting as peers to transmit data to a remote rendezvous node. In these scenarios, we will provide an analytical model in Sections 5.2.4 and 5.2.5 to predict TCP throughput in supporting IoT/M2M traffic over a satellite RA channel affected by segment losses due to collisions.

According to ETSI and IETF, CoAP is the reference protocol for M2M. Another IoT/M2M protocol, MQTT, developed by IBM in 1999, is the main opponent to CoAP, with a large history of successful deployments. The two protocols have several differences. For instance: (*i*) MQTT is TCP-based, instead CoAP is UDP-based; (*ii*) MQTT has (virtually) no limits on the payload size, while CoAP is limited to 1280 [B]; (*iii*) MQTT is based on a PUB/SUB paradigm, while CoAP provides a request/response interaction model and is REST-oriented (Representational State Transfer). CoAP has been designed with constrained devices in mind, and this explains the lightweight transport protocol in use and the limitation on the payload size, among others.

MQTT has been selected as reference protocol in this work because it is more common in real installations. Furthermore, the PUB/SUB paradigm implemented by MQTT foresees the use of a broker node (as already considered in the architecture in Figure 5.9), which acts similarly to a Split-TCP PEP, widely used in satellite networks.

We suppose that no data is lost (reliable delivery) in the local WSN among the IoT/M2M nodes and the local rendezvous node. The subscribers receive data from the rendezvous node via an RA satellite link. An MQTT-like application TCP-based protocol is assumed at the application layer, ensuring a reliable delivery of data from rendezvous nodes to subscribers. The only different assumption is that the underlining TCP connection is open only at the very first data transmission, to avoid a continuous 3WHS flow between a rendezvous node and one or more subscribers for the transmission of each data burst. The performance metric under consideration is the throughput at transport layer that corresponds to the throughput experienced by the application itself, apart from the overhead introduced at the application layer.

5.2.3.2 Satellite system

Let us consider a satellite system and a finite number N of RCSTs, sending data via a satellite RA return link. The RA protocol is CRDSA++. Time is slotted and each superframe contains one or more frames. In our configuration, an RA block spans the whole superframe and is composed of n time-slots. Each RCST can exploit any transmission opportunity to send dataover the return channel. In this work, each transmission opportunity can be used with probability equal to 1, if any data is ready for transmission. At layer 2, the Return Link Encapsulation (RLE) protocol encapsulates the datagrams coming from the network layer. RLE handles *encapsulation*, *fragmentation* and *frame packing* procedures. According to the upper-layer datagrams size, an RLE data unit can carry: a fraction of a datagram, the whole datagram or more than one datagram, according to the respective lengths. Let r denote the length of an RLE data unit measured in IP datagrams that can be packet together, while $f \geq 1$ is defined as the number of IP datagrams that are lost in a single MAC-layer collision event on the RA channel. In case of a collision, $f = \lceil r \rceil$ IP datagrams must be retransmitted. We assume no IP fragmentation in the following, so that one IP datagram carries a whole TCP segment.

We assume that the RCSTs are power balanced[9]. It is worth remarking that power-balancing is the worst condition RCSTs can face when using RA schemes, because the power unbalancing improves the benefits offered by the *capture effect*, thus improving the performance of SIC.

The DVB-RCS2 standard defines two burst lengths[10] (536 and 1616 symbols, respectively) to transport user data that can be transmitted by means of a rich set of WFs, so as to operate in different channel conditions and to support various traffic types. In more detail, the redundancy added by channel coding schemes as well as the modulation order can be tuned according to the channel conditions: DVB-RCS2 provides several predefined configurations, or

[9] In the reference scenario, RCSTs are uniformly distributed within the same beam. We assume that an uplink power control scheme is implemented, so that the power of the signal received from each RCST is approximately the same.

[10] Note that DVB-RCS2 also defines two additional burst lengths (664 and 262 symbols, respectively) transported by WF 1 and WF 2, which are however used only for control messages. Hence, they are not considered in this work.

Table 5.3: Details on the DVB-RCS2 WFs in use

WF ID	**Burst length**	**Payload length**	**Mapping scheme**	**Code Rate**
3	536	38 [B], 456 [symbols]	QPSK	1/3
14	1616	188 [B], 1504 [symbols]	QPSK	1/2

WFs, in order to satisfy reference Bit Error Rate (BER) figures. As a result, the time-slots of an RA block can accommodate different amount of data: a burst is defined as the amount of data fitting a time-slot. This feature can be exploited to configure DVB-RCS2 devices so as to handle different traffic types, such as small-payload messages characterising M2M-like traffic, or conversely larger ones. In the present work, WF 14 and WF 3 configurations are selected (see Table 5.3 for more details about these WFs). Apart from the chosen WFs, all the other parameters are equal in the scenario under consideration. The RA block of WF 14 is composed of time-slots, each carrying 188 bytes; the RA block of WF 3 is composed of time-slots, each carrying 38 bytes. Note that WF 3 is one of the most robust and used waveforms, relying on QPSK 1/3, while WF 14 uses QPSK 1/2. The choice of these two WFs stems from the fact that the corresponding volume of data being transported can be considered as representative of typical M2M message lengths.

5.2.3.3 TCP segment size and fragmentation at MAC layer

Two different MSSs are under consideration: 23 and 173 bytes, meant to exactly fit the payload size provided by WF 3 and by WF 14, respectively. IoT/M2M terminals generate low data-rates, thus the MSSs here considered are enough to periodically transmit small amount of data. Since we are dealing with small IoT/M2M payloads, Robust Header Compression (ROHC) is used to reduce the impact of TCP/IP overhead. ROHC is applied prior to transmitting data on the satellite return link. The payload size of the data stream is assumed to be constant, thus the 52 bytes of TCP/IP headers[11] reduce to 7 bytes for DATA segments and to 6 bytes for ACK segments[12].

[11] 40 bytes for TCP/IP headers plus 12 bytes for TCP options.

[12] See Section 4.4 of RFC 6846.

Table 5.4: Fragmentation at MAC layer

MSS + headers [bytes]	**WF ID**	$r = \frac{RLE\,payl.(WF)}{MSS+head.}$	$f = \lceil r \rceil$
23 + 15	3	1	1
23 + 15	14	5.7	6
173 + 15	3	0.175	1
173 + 15	14	1	1

The compression ratio for DATA is larger than 85%, thus effectively reducing the overhead.

Several scenarios are possible, depending on the chosen MSS and WF, because of the fragmentation that can take place at MAC layer prior to transmitting a TCP segment, as described in Table 5.4, where we consider that 15 bytes of headers (comprised TCP/IP headers plus layer-2 overhead) have to be summed to MSS. Three scenarios are possible, depending on the time-slot payload size:

1. the MSS has the same length ($r = 1$): one TCP segment exactly fits one MAC burst, thus a collision event causes a single segment loss, i.e., $f = 1$;
2. the MSS has a larger length ($r < 1$): one TCP segment fits into several MAC bursts. A collision event causes a single segment loss, because losing a fraction of the segment is equivalent to lose the whole segment, i.e., $f = 1$;
3. the MSS has a smaller length ($r > 1$): several segments fit into a single MAC burst, thus a collision event causes $f = \lceil r \rceil > 1$ segment losses.

The latter case is discussed separately because it violates the hypothesis of uncorrelated losses in Section 5.2.4. Moreover, we will show in Section 5.2.6 that this configuration has a poor performance, so that any configurations leading to $f > 1$ should be avoided, to reduce the possible waste of costly resources.

5.2.3.4 Interactions between TCP and the underlining RA scheme

The traffic arrival process for each RCST is driven by a sustained and continuous load at application layer, because each rendezvous node continuously collects IoT/M2M data from the WSN and sends them to the remote subscribers. The DVB-RCS2 standard specifies a layer-2 load control mechanism (see Section 5.2.6.3) aimed at keeping the system running at a target operating point G_T. However, in this work, the actual operating point G is determined by the TCP congestion control algorithm because it already provides a load control mechanism, even if it is meant for avoiding congestion events and not for dealing with RA collisions; because of this, the DVB-RCS2 load control algorithm is not adopted. Every time a collision event causes a burst loss, f TCP segments are lost, leading to the retransmission of the segments and to a reduction of the TCP sending rate.

TCP NewReno deals with segment losses relying on: (i) the use of the FR mechanism; (ii) the RTO event, triggered if FR is not successful or entering FR is not possible. CWND is, then, set to a lower value and the Slow Start Threshold (SST) is set accordingly. In both cases, the sending rate is reduced to counteract further loss events. In this work, the satellite return/forward links are supposed operating in clear-sky conditions[13] in order to focus only on segment loss events caused by collisions (ACKs are assumed to be always correctly received). No ARQ algorithm is in use at MAC layer. Each TCP segment sent to the lower layers (i.e., network and MAC layers) is queued into a large finite buffer, so that buffer losses can be neglected w.r.t. collision losses.

This paper analyses the capability of TCP to self-regulate its transmission rate; given N TCP flows belonging to N distinct RCSTs, the resulting offered MAC load per RCST is given by $\lambda = G/N$, i.e., the TCP flows contribute in achieving the maximum throughput at a load level G that depends on the dynamics of the congestion control algorithm: this aspect will be deepened in Section 5.2.6.3.

[13]DVB-RCS2 offers adaptive MODCOD (Modulation and Coding) to fulfill a target BER requirement. A quasi-error-free link assumption can be made by carefully choosing the target BER.

5.2.4 Analytical Model of TCP NewReno

This Section presents some analytical refinements to the throughput estimation model for TCP NewReno on top of RA. The throughput estimation models anticipated in Section 5.2.2.1 do not accurately match the empirical results of the simulation campaign, as shown in Section 5.2.6, thus motivating the refinements resented here. We stress that the throughput estimation model proposed in this Section is meant for TCP over an RA satellite link, and it has not been tested under different conditions. In this analysis, no assumptions are made on the specific RA protocol in use. Furthermore, given the IoT/M2M traffic profile under consideration, different TCP flavors may show basically the same performance level, because of limited CWND size and reduced available bandwidth, the latter being a typical case of satellite RA links.

The new model presented in the following, namely *NewRenoSAT*, is firstly developed in Section 5.2.4.1 assuming that all loss events are identified by TD ACKs, and then in Section 5.2.4.2 handling both TD ACKs and RTOs (the so-called *full model*).

5.2.4.1 Throughput model if loss indications are TD ACKs only

In this Section, we derive the model for estimating the TCP throughput in absence of RTOs, thus only considering CA and FR phases.

The throughput T of a TCP flow can be estimated by analysing a TCP cycle, an epoch of the renewal process. Let S_i^{CA} and S_i^{FR} be the number of segments successfully transmitted during the CA and FR phases of the i-th cycle, respectively, and S_i^{CAFR} the number of segments sent during a whole cycle, here denoted as Congestion Avoidance - Fast Retransmit / Fast Recovery (CAFR):

$$S_i^{CAFR} = S_i^{CA} + S_i^{FR}. \tag{5.3}$$

Let D_i^{CA} and D_i^{FR} denote the time duration of CA and FR periods and D_i^{CAFR} the duration of the i-th CAFR cycle:

$$D_i^{CAFR} = D_i^{CA} + D_i^{FR}. \tag{5.4}$$

The average throughput of a flow is given by (5.2). During the CA phase,

the receiver sends one ACK every b segments it receives (*delayed ACK* feature[14]), causing the CWND to increase linearly with a slope of $1/b$ segments per round, until the first segment loss occurs. Let us denote by α_i the first lost segment in the i-th cycle and by X_i the round where this loss occurs. The total number of segments sent in i-th cycle is $S_i^{CAFR} = \alpha_i + \gamma_i$, where γ_i is the number of segments sent between the first loss α_i and the last one in the drop window W_i. A *drop window* is defined as a CWND where a loss event has occurred. S_i^{CAFR} is defined as in Eq. (2) in [51]:

$$S^{CAFR} = \mathbb{E}[\alpha] + \mathbb{E}[\gamma], \tag{5.5}$$

where $S^{CAFR} = \mathbb{E}[S_i^{CAFR}]$. The expected number of segments sent in a cycle, having k rounds, up to α_i, is given in [52] as:

$$\mathbb{E}[\alpha] = \sum_{k=1}^{\infty} k(1-p)^{k-1}p = \frac{1}{p}, \tag{5.6}$$

where p represents the average loss event rate. A *loss event* is said to happen when one or more losses occur in a CWND, thus triggering the entering in FR (or triggering a RTO, as in Section 5.2.4.2). The first segment loss in a CWND marks the beginning of a *loss event*. The condition for entering the FR phase is to successfully deliver at least three segments in the drop window W_i. Hence, let us assume $\delta_i \geq 1$ losses with rate q over the remaining $(W_i - 3)$ segments, where q corresponds to the average SLR. The number of losses δ_i follows a binomial distribution over a drop window; in particular, let us consider the probability distribution of δ_i losses over $(W_i - 3)$ segments, conditioned on $\delta \geq 1$, as:

$$\begin{aligned} &Prob\{\delta = j \mid \delta \geq 1, W_i > 3\} = B[W_i - 3, j] \\ &= \binom{W_i - 4}{j-1}(1-q)^{W_i-3-j}q^{j-1}, j \in [1, W_i - 3]. \end{aligned} \tag{5.7}$$

Analogously to (5.6), the average number of segments sent between two consecutive losses is $1/q$. Therefore, if δ_i losses are assumed, the average number

[14] As RFC 1122 suggests, the amount of traffic from the receiver to the sender should be reduced by sending a single ACK every b segments.

of segments among δ_i losses in the same CWND is $(\delta_i - 1)/q$. The expected value of γ can be calculated as follows:

$$\begin{aligned}\mathbb{E}[\gamma] = & \ \mathbb{E}\left[\sum_{j=1}^{W_i-3} \frac{(j-1)}{q}\ (q\,B[W_i-3, j]\,)\right] \\ & = q\,(\mathbb{E}[W]-4),\end{aligned} \tag{5.8}$$

where $\mathbb{E}[W]$ is the expected value of the CWND size, under the assumption of steady state, and $(q\,B[W_i-3, j])$ is the joint probability of the first loss α_i and of the other $(j-1)$ losses in the drop window. Note that $\mathbb{E}[\gamma]$ is defined only in presence of the losses. If just a single segment loss occurs, then $\mathbb{E}[\gamma] = 0$; otherwise, if two or more segment losses occur, then $\mathbb{E}[\gamma] > 0$. When $q \to 1$, $\mathbb{E}[\gamma] \to (\mathbb{E}[W]-4)$, i.e., the number of segments sent in a drop window after the first three segments allowing for TD ACKs and the first loss α; conversely, when $q \to 0$, $\mathbb{E}[\gamma] \to 0$.

In order to derive the average number of rounds in a cycle, namely $\mathbb{E}[X]$, and the average CWND size, let us consider the evolution of CWND as a function of the number of rounds. According to [52] and [64], we have:

$$W_i = \frac{W_{i-1}}{2} + \frac{X_i}{b} - 1, \tag{5.9}$$

thus we can write the following relation, at regime:

$$\mathbb{E}[X] = b\left(\frac{\mathbb{E}[W]}{2} + 1\right). \tag{5.10}$$

The expected number of segments sent during CA is determined in [64] as:

$$S^{CA} = \frac{\mathbb{E}[X]}{4}\,3\,\mathbb{E}[W] + \mathbb{E}[\beta], \tag{5.11}$$

where β_i is the number of segments sent in the last round of the i-th cycle. $\mathbb{E}[\beta] \approx \mathbb{E}[W]/2$ because we assume β to be uniformly distributed in $[1, W_i - 1]$.

The number of segments $S_{i,k}$ sent at each round k of the *i-th* FR cycle, considering the Partial Window Deflation[15] mechanism, can be derived from [69] as:

$$S_{i,k} = \max\Big(0,\ W_i/2 - \delta_i + k - 1\Big),\ 1 \le k \le \delta_i. \tag{5.12}$$

[15]RFC 6582: The NewReno modification to TCP's fast recovery algorithm.

The entering in an FR phase is due to the reception of a TD. When a partial ACK is received, CWND is decreased by the amount of data acknowledged and increased of one segment size. Therefore $S_{i,k+1} = S_{i,k} + 1$. $S_{i,k}$ has a maximum value for $k = \delta_i$, then $\max(S_{i,k}) = S_{i,\delta_i} = W_i/2 - 1$. Thus, the expected total number of segments sent in an FR cycle comes from (5.12) as follows:

$$\begin{aligned} S^{FR} &= \mathbb{E}\Big[\sum_{k=1}^{\delta_i} S_{i,k}\Big] \\ &= \begin{cases} \frac{1}{2}(\mathbb{E}[\delta]\,\mathbb{E}[W] - \mathbb{E}[\delta] - \mathbb{E}[\delta]^2), & \text{if } \mathbb{E}[\delta] < \mathbb{E}[W] \\ 0 & \text{if } \mathbb{E}[\delta] = \mathbb{E}[W]. \end{cases} \end{aligned} \tag{5.13}$$

In the case of a single loss, i.e., $\delta_i = 1$, $S^{FR} = \mathbb{E}[W]/2 - 1$. After the first loss, the remaining $(\delta_i - 1)$ segments are lost over $(W_i - 4)$ segments. Hence, we can write the following formula to characterize $\mathbb{E}[\delta]$, the expected value of the number of losses δ_i:

$$\mathbb{E}[\delta] = \mathbb{E}\left[\sum_{j=1}^{W_i-3} j\, B[W_i - 3, j]\right] \approx 1 + (\mathbb{E}[W] - 4)\, q. \tag{5.14}$$

Assuming that collisions on an RA channel are independent events and $f = 1$, then the segment losses are independent events, too. Instead, if $f > 1$, each collision in the channel causes a burst of segment losses of average length in the interval $[1, f]$, thus TCP segment losses cannot be considered as independent ones. Further details on this are in Section 5.2.6.1.

Using (5.11) and (5.13), we can express the expected number of segments sent during a CAFR cycle, i.e., the expected value of (5.3), as follows:

$$S^{CAFR} = S^{CA} + S^{FR}. \tag{5.15}$$

Then, the expected value of CWND, $\mathbb{E}[W]$, can be obtained by equating (5.5) and (5.15). In particular, we obtain a second order equation in $\mathbb{E}[W]$, whose only positive solution can be expressed as:

$$\mathbb{E}[W] \approx \Phi + \sqrt{\Phi^2 - \frac{60pq - 8p - 8}{p(8q + 3b)}}, \tag{5.16}$$

where $\Phi = \frac{22q-3b-4}{8q+3b}$.

Equation (5.16) explicitly takes b into account, differently from Eq. (14) in [51], where $b = 1$ is assumed. We will show in Section 5.2.6 that, because of the finer estimations of S^{FR} and $\mathbb{E}[\delta]$, the expected value of CWND in (5.16) is closer to the simulated one than the other theoretical models under consideration [51, 52, 53].

Each RTT is supposed as a r. v., whose value does not depend on CWND, as in [51, 52, 53]. We recall that:

$$D^{CA} = \mathbb{E}[X] + \mathbb{E}[D^{\beta}], \tag{5.17}$$

where D^{CA} is the average duration, in number of rounds, of the CA phase, including the time spent waiting for entering in FR. Moreover, D^{FR} is the average duration, in number of rounds, of the FR phase:

$$D^{FR} = \mathbb{E}[\delta]. \tag{5.18}$$

In fact, NewReno recovers one lost segment per round. Finally, $\mathbb{E}[D^{\beta}]$ is the expected time, in number of rounds, after the first loss and before entering FR; it can be approximated as $\mathbb{E}[D^{\beta}] \approx 1/2$. From (5.17) and (5.18), the expected duration of a cycle $D^{CAFR} = \mathbb{E}[D_i^{CAFR}]$ [s], results as:

$$D^{CAFR} = \mathbb{E}[RTT]\big(D^{CA} + D^{FR}\big). \tag{5.19}$$

Figure 5.10 shows the evolution of CWND during CA and FR phases and the duration of the aforementioned phases. A CAFR cycle ends with the end of the FR phase because of one or more segment losses. The duration of the last part of the CA phase is D^{β} RTTs: TD ACKs are received, prematurely ending the last round of the CA phase. The two segments lost in Figure 5.10 are recovered during the FR phase immediately following, which lasts $D^{FR} = 2$ RTTs. The CWND size is $(W_i/2 + 3)$ at the start of the FR phase and $W_i/2$ at the end of the FR phase.

Combining (5.5) and (5.19), we can finally write the throughput expression [segments/s], neglecting RTOs, as:

$$T_{noTO} = \frac{(\mathbb{E}[W] - 4)\, q + \frac{1}{p}}{\mathbb{E}[RTT]\left\{b(\frac{\mathbb{E}[W]}{2} + 1) + \frac{1}{2} + [1 + (\mathbb{E}[W] - 4)\, q]\right\}} \tag{5.20}$$

where $\mathbb{E}[W]$ is expressed in (5.16).

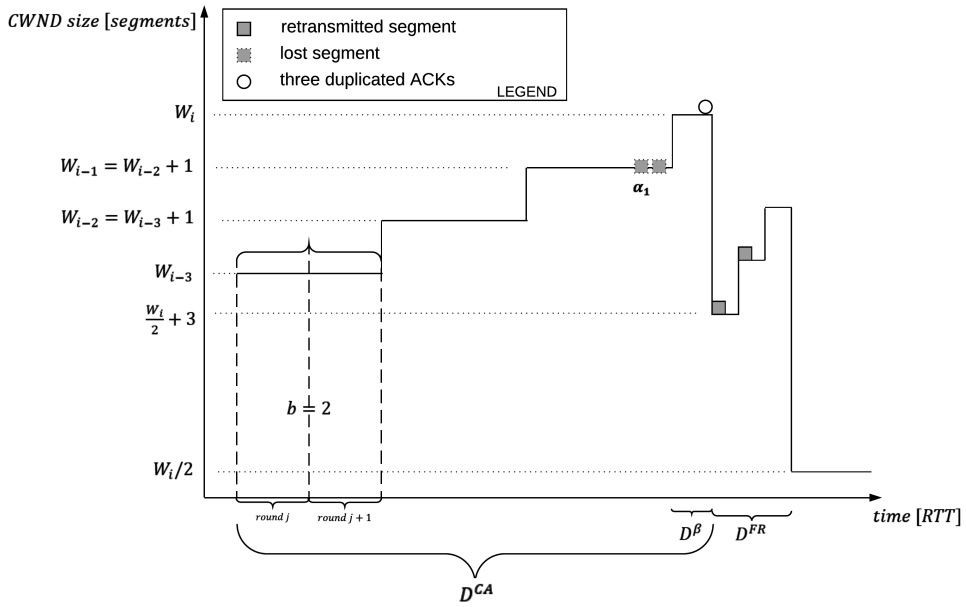


Figure 5.10: Evolution of CWND during CA and FR phases

5.2.4.2 Full Model

This section extends the model in Section 5.2.4.1 to include also RTOs as loss indications. Each cycle consists of either a CA or a CAFR period followed by a RTO with an SS period. TCP NewReno may experience a RTO either during CA or during FR, given that a loss event (L) occurred with probability p. The former transition occurs with probability $p_{TOCA} = P\{RTO \,|\, CA,\, L\}$, when TCP does not receive enough duplicate ACKs to trigger FR, while the latter transition occurs with probability $p_{TOFR} = P\{RTO \,|\, FR,\, L\}$, when retransmitted segments are lost during the FR phase. TCP can experience a RTO in the CA phase when more than $(W - 3)$ segments are lost in a drop window. The conditional probability p_{TOCA} is given by:

$$p_{TOCA} = \mathbb{E}\left[\sum_{j=W_i-2}^{W_i} \binom{W_i - 1}{j-1}(1-q)^{W_i-j} q^{j-1}\right]. \tag{5.21}$$

A RTO occurs during FR if any of the retransmitted segments is lost. This condition can be approximated by assuming that if a loss event occurs in FR, then also the retransmitted segment is lost, thus triggering a RTO [51]. For δ losses in the drop window, NewReno takes δ RTTs to recover the lost segments, during which it sends S^{FR} segments. The probability that the i-th segment is lost, given that the previous $(i-1)$ segments are successfully delivered, is $(1-p)^{i-1}p$. Therefore, according to [51], it follows that:

$$p_{TOFR} = \mathbb{E}\left[\sum_{j=1}^{W_i-3} B[W_i-3,j]\left[1-(1-p)^{j\frac{W_i}{2}}\right]\right]. \tag{5.22}$$

During a RTO, TCP does not send any segments. According to [53], the duration of a RTO period is given by:

$$D^{RTO} = TO\frac{1+\sum_{j=0}^{5} 2^j p^{j+1}}{1-p}, \tag{5.23}$$

where the initial value of the TO parameter of TCP is 2 or 3 [s], in recent implementations[16].

In the SS phase, the initial CWND size is one and it grows until the SST, $W/2$, is reached. Therefore, in the last round of SS, TCP transmits $\mathbb{E}[W]/2$ segments, on average, then enters the CA phase. The rate of increase of the CWND, when the *delayed ACKs* feature is in use, is $x = (1+\frac{1}{b})$ on an RTT basis [70]:

$$\begin{aligned} S^{SS} &= \mathbb{E}\left[x^0 + x^1 + ... + x^{\log_x \frac{W_i}{4}}\right] = \mathbb{E}\left[b\left(x\frac{W_i}{4}-1\right)\right] \\ &= b\left(x\frac{\mathbb{E}[W]}{4}-1\right). \end{aligned} \tag{5.24}$$

S^{SS} represents the expected value of the number of segments sent during the SS phase as in [51], where we consider the segments of the last round of the SS phase as being part of the CA phase. D^{SS} represents the duration of the

[16] As in RFC 6298, *Appendix A*.

SS phase as proposed in [51], but here explicitly considering b:

$$\begin{aligned} D^{SS} &= \mathbb{E}\left[RTT_i\left(\log_x \frac{W_i}{4}+1\right)\right] \\ &\approx \mathbb{E}[RTT]\left(\log_x \frac{\mathbb{E}[W]}{4}+1\right). \end{aligned} \tag{5.25}$$

Note that the expected value in (5.25) is applied to a non-linear expression. We have already assumed RTT_i and W_i as independent r. v.. Moreover, if the distribution of W is concentrated around its mean, as shown in Section 5.2.6, then the approximation on the derivation of the expected value in (5.25) is acceptable.

The final expression for the throughput estimation, when considering also RTOs, is given in (5.26). It is worth remarking that some parameters, for instance $\mathbb{E}[W]$ provided in Section 5.2.4.1, are used also here, but they only consider CA and FR phases, without taking SS into account, thus providing an approximated estimate. The formulation in (5.26) is presented in its whole expression, without any approximations. We will show in Section 5.2.6 that (5.20) represents a throughput estimation that shows a very small error, so that the improvement provided by (5.26) is negligible.

$$T_{Full} = \frac{(1-p_{TOFR}-p_{TOCA})(S^{CA}+S^{FR})+p_{TOCA}(S^{SS}+S^{CA})+p_{TOFR}(S^{SS}+S^{CA}+S^{FR})}{(1-p_{TOFR}-p_{TOCA})(D^{CA}+D^{FR})+p_{TOCA}(D^{SS}+D^{CA}+D^{RTO})+p_{TOFR}(D^{SS}+D^{CA}+D^{FR}+D^{RTO})}. \tag{5.26}$$

5.2.5 The Relation between Burst Loss Rate and Event Loss Rate: the BLR Model

In this Section, a simple but effective model is proposed in order to glue together the BLR value experienced at MAC layer and the p and q values at transport layer. BLR is the rate of losses due to collisions. In [10], an analytical recursive model is proposed to derive an upper bound for the throughput of 2-CRDSA, but, to the best of the authors' knowledge, no analytical model is available for estimating throughput and BLR of CRDSA++.

In the following, we use the term *collision* to refer to MAC bursts marked as lost after SIC, thus not recoverable in any way at the receiver. TCP segments are retransmitted if they are lost because of collisions. For the sake of simplicity, in the following we assume that one segment fits exactly into one

time-slot (case $r = 1$). We recall that buffer overflow effects are negligible because of the large buffer in use. In the simulation results presented in Section 5.2.6, each TCP retransmission corresponds to a collision in the channel. In our study and simulations, we neglect spurious TCP retransmissions.

In what follows, a very simple expression is provided, denoted as the *BLR model*, which links together q, p and BLR. This expression holds if no spurious retransmissions occur and $f = 1$. We recall that p is the rate of loss events, q corresponds to SLR and $\mathbb{E}[\delta]$ is the expected number of segments lost in a loss event. Each collision is responsible of a single segment loss, triggering a retransmission at transport layer. Thus, we can write:

$$q \equiv BLR \tag{5.27}$$

and, for a finite CWND size:

$$p = \frac{q}{\mathbb{E}[\delta]} \approx \frac{q}{1 + (\mathbb{E}[W] - 4)\, q} \approx \frac{BLR}{1 + (\mathbb{E}[W] - 4)\, BLR}. \tag{5.28}$$

Plugging (5.27) and (5.28) in (5.20) and (5.26), we obtain the throughput estimation as a function of BLR. Following this approach, it is sufficient to know the collision rate at MAC layer, i.e., BLR, to obtain q and p rates to derive a TCP throughput estimation. In Section 5.2.6.2, the *BLR model* is validated against simulation results.

5.2.6 Performance Evaluation

The performance evaluation shown here is based on the NS-3 simulator, implementing a DVB-S2/RCS2 satellite network by means of the modules described in [6]. The TCP flavor in use is NewReno. The available implementation in NS-3 is based on the SBS variant. Note that, as shown in [52], the performance of SBS is not significantly different from that of the Impatient variant in the conditions here under consideration, i.e., limited congestion window sizes and independent loss events. The main settings of the simulator are provided in Table 5.5. The results here intend to assess how changing WF and MSS values can alter the overall performance. The main aim of this study is in clarifying if TCP is a suitable choice as transport protocol over a long delay RA link and, if this is the case, which choice of parameters leads to the highest throughput.

Table 5.5: System settings

Name	**Value**
TCP flavor	NewReno
TCP MSSs	23 / 173 [B]
TCP/IP headers size	7 [B] (w/ ROHC)
Initial TO value	2 [s]
RA scheme	CRDSA++ (3 replicas)
RA blocks per superframe	1
RA block duration	13 [ms]
Time-slots per RA block, WF 14	64
Time-slots per RA block, WF 3	194
Bandwidth	8012820 [Hz]
Roll off	0.2
Carrier spacing	0.3 [Hz]
Nominal RTT	0.52 [s]

We recall that the G^* load, i.e., the normalized MAC load level achieving the highest throughput at MAC level, depends on the number of available time-slots per RA block [46], among other factors. Thus, while keeping the other parameters unmodified, the choice of a different waveform can change the number of available time-slots per RA block, shifting G^*. For instance, when 64 time-slots are available, the G^* value of CRDSA++ is ≈ 0.7 [bursts/time-slot]; when 194 time-slots are available, the G^* value of CRDSA++ is ≈ 0.78 [bursts/time-slot] [46]. The overall system has been tested at several increasing load levels, before, close to and after G^*. Two crucial aspects are characterized in this work: the MAC operating point $\hat{G}_T$ when NewReno is in use over an RA satellite channel, and the system stability; both issues are discussed in Section 5.2.6.3. To the best of the authors' knowledge, the characterization of $\hat{G}_T$ is not available in literature for CRDSA++, if we exclude our preliminary contribution in [24].

The focus in what follows is on normalized load levels $G \geq 0.45$, because,

up to this point, CRDSA++ throughput is almost equal to the offered load (i.e., the operating point is in the linear part) (see Figure 3.2); beyond this load level, the system works in the quasi-linear part up to the G^* point, and then the MAC throughput dramatically decreases because of a too large BLR. If a DVB-RCS2 satellite system with ARQ enabled (typical setting) works at $G > G^*$ and no load control algorithm is used, the system may experience instability [48].

5.2.6.1 Throughput validation

The NewRenoSAT model described in Section 5.2.4 is here validated against simulation results and compared with other approaches in literature (see Section 5.2.2.1): (*i*) the contribution in [53], then amended in [64] (here denoted as PFTK model); (*ii*) the contribution in [51] (here denoted as PWM model); (*iii*) the contribution in [52] (here denoted as DAKH model). In the following, the provided throughput estimations are obtained by using simulated p and q values because there is no analytical model in literature able to characterize the layer-2 losses when CRDSA++ is in use.

Accuracy of the CWND estimation A critical value to be computed, when estimating the TCP throughput, is the CWND size. While just the expected value is used for the throughput estimation, the CWND exhibits a range of different values during the whole TCP connection lifetime. Figure 5.11a shows the distribution of CWND obtained from simulations, when WF 14 and MSS = 173 bytes are selected; in the following, this scenario is used as exemplary case. As the load increases (i.e., when the number of RCSTs increases), $\mathbb{E}[W]$ decreases, while the distribution keeps the same shape. A larger G value means higher load on the RA channel and then higher BLRs. Each time two or more RCSTs are involved in a collision, their CWND size is shrunk to a lower value, as in Figure 5.10 after the two lost segments. Therefore, $\mathbb{E}[W]$ decreases as G increases. In Table 5.6, the simulation results are compared with the NewRenoSAT model estimations, showing a close match.

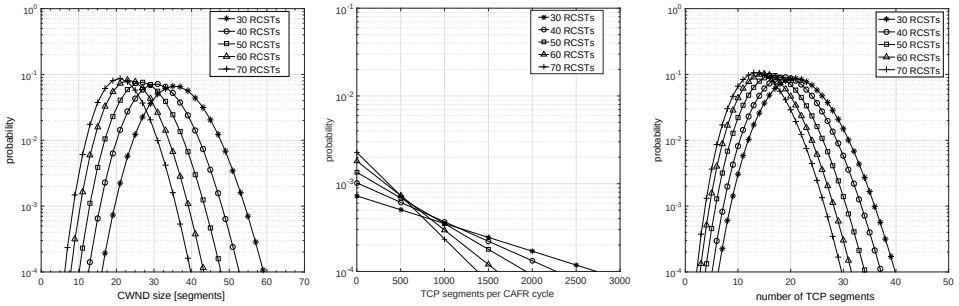
The average value of the CWND has a direct impact on the number of segments sent in a CAFR phase, as visible in (5.15). The distribution of the number of segment sent per CAFR phase is shown in Figure 5.11b: this

Table 5.6: NewRenoSAT estimated values vs. simulation results
WF 14, MSS = 173 bytes, 64 time-slots per RA block

#RCSTs	$\mathbb{E}[W]$		S^{CA} [seg.], D^{CA} [s]		S^{FR} [seg.], D^{FR} [s]	
	sim.	model	sim.	model	sim.	model
30	40.15	41.8	1367.6, 25.5	1394, 28.4	19.3, 0.62	20.5, 0.66
40	33.4	34.8	1003.7, 21.4	978, 22.4	15.6, 0.58	17.1, 0.63
50	28.5	29.6	721.4, 17.9	716.6, 18.6	13.2, 0.58	14.4, 0.61
60	26.1	26	552.1, 15.6	559.2, 16.5	11.7, 0.58	12.6, 0.61
70	21.6	21.9	433.3, 13.9	406.1, 14.2	10.4, 0.58	10.6, 0.62

is a geometric distribution, as pointed out in [53]. In a CAFR cycle, the sending rate varies if the connection is in the CA phase or in the FR phase: to underline this, Figure 5.11c shows the distribution of the number of segments sent in FR only. The distribution in the FR phase (in Fig. 5.11c) is different from the one in the CAFR phase (in Fig. 5.11b): the CA phase completely dominates the dynamics of a CAFR cycle, and it exhibits an approximate geometric distribution, while the FR phase has quite a negligible impact on the overall distribution, thus the approximation of a geometric distribution still stands. Table 5.6 also shows how the NewRenoSAT model is able to accurately estimate the average number of segments in CA and FR phases, along with their average durations.

Average number of segment losses per loss event A central part of this work is the characterization of the loss process experienced on the RA channel when CRDSA++ is in use. Equation (5.14) provides the analytical expression of $\mathbb{E}[\delta]$, the expected number of losses per loss event. It ranges from 1 to 1.2 in our simulation results, as shown in Table 5.7, if we exclude the case of WF 14 and MSS = 23 bytes, later discussed. It means that, on average, 1 to 1.2 segments are lost per drop window: the same number of segments is retransmitted in the FR phase, which follows the loss event. Because of the reduced available bandwidth per RCST, the average CWND size is small and, on average, a single loss is experienced per drop window if q is small. Instead, a strong correlation effect is present in the scenario WF 14 and MSS



(a) PMF of the CWND size

(b) PMF of the number of segments sent per CAFR cycle

(c) PMF of the number of segments sent per FR phase

Figure 5.11: Simulation results of the scenario WF 14, MSS = 173 bytes, 64 time-slots per RA block (PMF stands for Probability Mass Function)

= 23 bytes: a single time-slot carries up to $r = 5.7$ segments, thus a single collision causes a burst of $f = 6$ losses at transport layer. Note that r is the maximum number of whole or partial TCP segments into a single time-slot, and not the actual one, because the RLE protocol always tries to ensure the largest occupation of the time-slot, but it is limited by the number of data units actually pending for transmission. It means that, if less than f data units are in the queue, a smaller number will be transmitted in a single time-slot, thus sub-utilizing the available resources. Moving to the performance evaluation, in this case a single collision represents a very stressful event for a connection, because of the burst of losses experienced at TCP layer. This scenario exhibits very low performance because of this effect, when compared with the scenario WF 14 and MSS = 173 bytes and, generally, its performance are worse than any other configuration with $f = 1$. Because of this, we are not intended to the study of the TCP behaviour under the hypothesis of bursty losses due to encapsulation effects (i.e., $f > 1$). For the same reason, Table 5.7 does not show the $\mathbb{E}[\delta]$ value, whose analytical expression is valid only under the hypothesis of independent losses.

Comparison among different approaches for estimating the TCP throughput In Figures 5.12, 5.13 and 5.14, we compare the different ap-

proaches for the analysis of TCP throughput over a RA satellite link using the CRDSA++ protocol. The throughput estimation relies on the use of p, q and $\mathbb{E}[RTT]$ values from simulations, as in Table 5.7.

The relative error $|\eta = 1 - \frac{T_{est}}{T_{sim}}|$ of the estimations is shown, where T_{est} is the throughput resulting by the analytical models under consideration [51, 52, 53] and NewRenoSAT, and T_{sim} is the throughput resulting from the simulations. The results here shown are obtained with extensive Montecarlo simulation runs: more than 1 million sent segments per scenario under consideration, with a narrow confidence interval. η is computed for every combination of WF and MSS in use, except for the scenario WF 14 and MSS = 23 bytes ($f > 1$), as explained before. Let us recall that $b = 2$ (delayed ACKs) is used in the simulations in this work; the PMW model does not account for it, so the relative error of this analytical model is larger when compared with the others under consideration.

Figure 5.12 shows the value of η for the scenario WF 3 and MSS = 23 bytes: 194 time-slots are available for transmission and an RCST can use one time-slot per RA block. A segment fits exactly into one MAC burst ($r = 1$). The NewRenoSAT model proposed in this work show the lowest η value (i.e., the best results), if we except the case with the largest number of RCSTs (i.e., the largest load); the DAKH model and the PFTK one are quite accurate, too, but the latter has a η larger than 0.15 for higher loads and the former tends to overestimate the throughput at medium loads.

In Figure 5.13 the relative error for the scenario WF 3 and MSS = 173 bytes is shown. In this scenario, a time-slot carries a fraction of the segment: in fact, $r = 0.175$, so that 5.7 time-slots are needed to carry a single segment, which means that the sixth time-slot carries the last fraction of a segment and the initial fraction of the next one. A collision involving the sixth time-slot causes the loss of two TCP segments, thus triggering two retransmissions for a single collision event. This happens with probability equal to 1/6: because of this, the error on the estimation is larger than the one in Figures 5.12 and 5.14, for each model. In this scenario, there is some correlation among losses, so the hypothesis of complete independence among segment losses is not verified. Anyway, the correlation is limited, so the simulation results show a good match with NewRenoSAT and DAKH analytical models anyway.

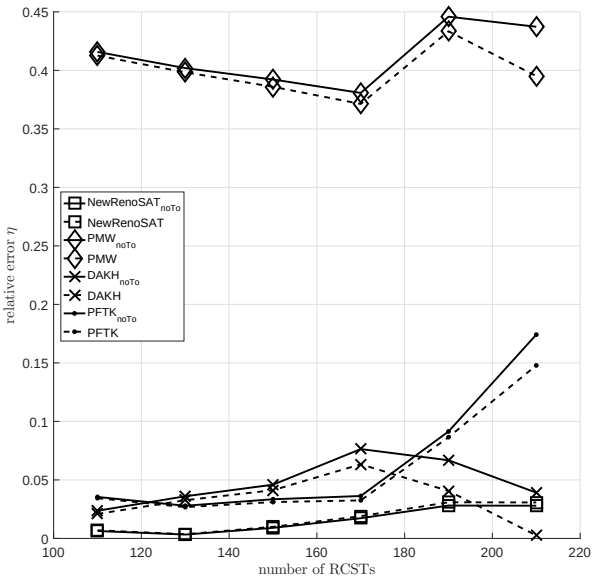


Figure 5.12: Comparison (relative error) between simulations and theoretical approaches for TCP NewReno throughput over CRDSA++ for WF 3, MSS = 23 bytes and 194 time-slots per RA block

Figure 5.14 provides the throughput estimation for the scenario WF 14 and MSS = 173 bytes: one segment fits exactly into a single MAC burst ($r = 1$). NewRenoSAT and DAKH models achieve the best results, even if DAKH shows an average relative error that is larger at high loads. The PFTK model exhibits a larger error, instead; in fact, it does not take the FR mechanism into account.

Retransmission TimeOuts In this Section, the measure ξ is introduced to quantify the rate of RTOs in the simulations:

$$\xi = \frac{\mathbb{E}[\#RTOs]}{T_s/\mathbb{E}[RTT]}. \tag{5.29}$$

$\mathbb{E}[\#RTOs]$ is the average number of RTOs per connection and T_s is the simulation length in seconds, where $T_s/\mathbb{E}[RTT]$ is the number of rounds in each simulation. Therefore, ξ represents the probability that a RTO occurs in a

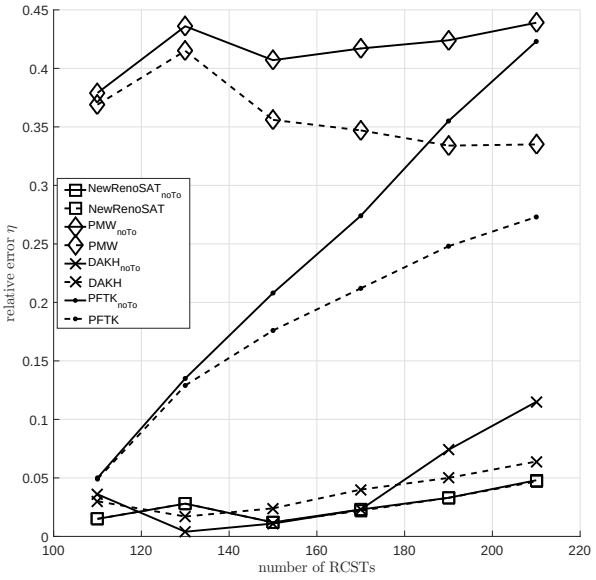


Figure 5.13: Comparison (relative error) between simulations and theoretical approaches for TCP NewReno throughput over CRDSA++ for WF 3, MSS = 173 bytes and 194 time-slots per RA block

round. The numerical values of ξ per scenario under consideration are shown in the last column of Table 5.7.

We observe two effects worth of mentioning: if $r = 1$, ξ has a substantially lower value with respect to the case where $r \neq 1$; then, MAC fragmentation and packing significantly impact on TCP performance. The second effect here discussed is related to the load level: the lowest simulated load level exhibits a ξ value larger than that at higher ones, which is unexpected, at a first glance. At high loads, the collision rate is responsible of sustained segment losses, thus the ξ value reflects this. At medium loads, the collision rate is responsible of moderate losses, thus ξ shows a lower value. In fact, the FR mechanism is particularly efficient in presence of a moderate loss rate, and time-consuming RTOs are avoided. Instead, at low loads, another effect is present, which makes ξ larger than expected: the CWND overshooting problem. A collision is a very rare event at low loads, therefore the overshooting problem is possible: a CWND that increases slowly but continuously over time and whose value

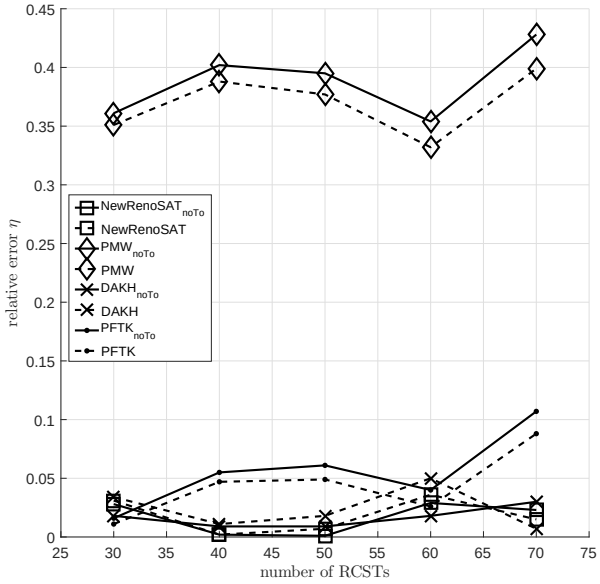


Figure 5.14: Comparison (relative error) between simulations and theoretical approaches for TCP NewReno throughput over CRDSA++ for WF14, MSS = 173 bytes and 64 time-slots per RA block

goes over the Bandwidth-Delay product increases the probability of spurious RTOs.

5.2.6.2 Validating the BLR model

The *BLR model* discussed in Section 5.2.5 is now validated against simulation results. The throughput estimation here presented relies on the use of BLR and $\mathbb{E}[RTT]$ values taken from simulations, as in Table 5.7. The scenario WF 14 and MSS = 173 bytes has been chosen because of the very low rate of spurious retransmissions.

Table 5.8 shows the relative error η when comparing the simulation results with the estimations provided by (5.20) and (5.26) in the first two columns. Further to this, the accuracy of the *BLR model* is visible in the last two columns, i.e., when plugging (5.27) and (5.28) in (5.20) and (5.26). The *BLR*

Table 5.7: Simulation results: the scenarios are described in the first column. $\mathbb{E}[RTT]$ is measured in seconds, $Thr.$ is the average simulated throughput [kbps] per TCP connection

WF, MSS	$\#RCSTs$	BLR	r	f	q	p	$\mathbb{E}[\delta]$	$\mathbb{E}[RTT]$	$Thr.$	ξ
3, 23	110	1.1E-4	1	1	1.44E-3	4.5E-4	1.07	0.69	12.50	5.5E-4
3, 23	130	2E-4	1	1	1.70E-3	4.86E-4	1.08	0.68	11.90	3E-5
3, 23	150	5.5E-4	1	1	2.80E-3	6.68E-4	1.11	0.62	9.60	6.2E-5
3, 23	170	9.2E-4	1	1	5.13E-3	9.86E-4	1.16	0.61	7.84	5E-4
3, 23	190	1.1E-3	1	1	7.94E-3	1.20E-3	1.20	0.60	6.86	2.5E-3
3, 23	210	2.9E-3	1	1	1.1E-2	6.3E-3	1.20	0.59	2.90	4E-3
3, 173	110	1.2E-4	0.175	1	4.1E-3	1.5E-3	1.09	2.40	11.94	1E-1
3, 173	130	3.5E-4	0.175	1	8.66E-3	3.5E-3	1.14	1.50	11.90	9E-2
3, 173	150	1.4E-3	0.175	1	2.35E-2	1.1E-2	1.16	0.92	10	7E-2
3, 173	170	2.8E-3	0.175	1	2.94E-2	1.72E-2	1.17	0.77	8.50	5E-2
3, 173	190	4.4E-3	0.175	1	3.3E-2	2.7E-2	1.18	0.72	7.30	6E-2
3, 173	210	6.16E-3	0.175	1	3.5E-2	3.5E-2	1.20	0.69	6.37	7E-2
14, 23	30	2.27E-3	5.7	6	1.3E-2	3.7E-4	-	1.28	10.5	5E-2
14, 23	40	2E-3	5.7	6	1.38E-2	3.8E-4	-	1.275	10.48	4E-2
14, 23	50	2.32E-3	5.7	6	1.64E-2	4.2E-4	-	1.23	10.1	4E-2
14, 23	60	2E-3	5.7	6	1.71E-2	4.3E-4	-	1.19	9.93	4E-2
14, 23	70	1.9E-3	5.7	6	1.76E-2	4.7E-4	-	1.15	9.72	4E-2
14, 173	30	6.41E-4	1	1	8.52E-4	7.45E-4	1.03	0.64	65.75	5.3E-4
14, 173	40	9.26E-4	1	1	1.34E-3	1E-3	1.04	0.60	58.50	1E-4
14, 173	50	1.31E-3	1	1	1.75E-3	1.37E-3	1.04	0.58	51.51	1.4E-4
14, 173	60	1.76E-3	1	1	2.30E-3	1.79E-3	1.05	0.58	45.20	3E-4
14, 173	70	2.3E-3	1	1	3.5E-3	2.28E-3	1.07	0.58	38.11	4.6E-4

model provides good precision: in fact, the larger estimation error is $\leq 7\%$. A consideration is here in order to motivate the need of the *BLR model*: while it can be practical to estimate BLR in real systems, the same cannot be said for p and q when using TCP. Thus, this model offers a simple and elegant way, in the authors' opinion, for estimating the TCP throughput over an RA satellite link.

5.2.6.3 Stability

A critical issue to be taken into account, when dealing with RA protocols, is system stability. CRDSA++, as other RA protocols, has an optimal working

point, namely G^*, which exhibits the maximum throughput offered by the MAC protocol, namely T^*. If the system is forced to work at loads $G > G^*$, instability may occur and proper countermeasures are needed, as analysed in [48].

DVB-RCS2 standard encompasses a normative load control algorithm, aimed at keeping the system at a target operating point G_T, where $G_T \leq G^*$. This load control algorithm is not used in our simulator because the work in [15] shows that the aforementioned algorithm needs to be tuned in accordance with each scenario under consideration, and this can be not trivial at all; furthermore, a more complex algorithm would be necessary to target realistic QoS requirements. On the contrary, the use of TCP ensures the stability at every load level under consideration, as discussed later in this part, and does not require any tuning. The DVB-RCS2 load control algorithm can limit the magnitude of the load oscillations around G_T according to two different strategies [15]: by reducing the number N of RCSTs or by shifting G_T to a lower value. These oscillations can be responsible for pushing the instantaneous offered load G beyond G^*, thus leading the system towards the instability region; hence, controlling these oscillations is of primary importance. Both strategies offer disadvantages: in an IoT/M2M scenario, a large population of RCSTs is common, so that reducing N could be unfeasible. Instead, shifting G_T towards a lower value can be a better solution in our scenario, but the use of a centralized NCC, which periodically monitors the aggregate load, is needed. NCC is responsible for choosing the G_T value according to the instantaneous G. Figure 5.15, later described, shows that the use of TCP removes the need of a centralized control: TCP can control G_T. Hence, TCP ensures the stability, without requiring any layer-2 control mechanisms that can improperly interact with the TCP congestion control algorithm.

Figure 5.15 shows the number of RCSTs versus the normalized MAC throughput and normalized MAC offered load, when using TCP on top of WFs 14 and 3. We recall that WF 3 offers a greater number of time-slots carrying small payloads, while WF 14 offers a lower number of time-slots carrying larger payloads. Three load intervals can be read in Figure 5.15, w.r.t. the number of RCSTs: the first one, at low loads, up to ≈ 100 RCSTs, where WF 14 offers a larger normalized throughput; the second one, at medium loads,

from ≈ 100 to ≈ 350 RCSTs, where WF 3 outperforms WF 14 thanks to the greater number of time-slots; finally, the third interval, at larger loads, from ≈ 350 RCSTs on, where the throughput offered by the two WFs is almost comparable. A consideration is here in order: when using WF 3, TCP behaves in a greedy way, showing a clear load peak when ≈ 150 RCSTs are present, then the load decreases as the number of RCSTs increases. On the contrary, WF 14 shows a more balanced behavior, and a peak is not recognizable. The key difference is the number of available time-slots, which in turns determines the number of active RCSTs per RA block. The number of active RCSTs per RA block determines the SLR that each RCST experiences, because, for a given load G, the corresponding BLR can be estimated, and the BLR model in Section 5.2.5 can be used to relate BLR and SLR.

Let us call $\hat{G}_T$ the working point for TCP NewReno on top of CRDSA++ protocol. According to the y-axis values in Figure 5.15, we can see that the following ranges are possible for the working point: (*i*) $\hat{G}_T \in [0.45, 0.55]$ for TCP over WF 14 (with 64 available time-slots); (*ii*) $\hat{G}_T \in [0.58, 0.63]$ for TCP over WF 3 (with 194 available time-slots). When CRDSA++ (3 replicas) is in use and assuming no flow control algorithms, (i.e., TCP is not in use, as well), $G^* \approx 0.7$, when 64 time-slots are available per RA block [46], and $G^* \approx 0.78$, when 194 time-slots are available per RA block [46]. Thus, as Figure 5.15 shows, $\hat{G}_T \leq G^*$: the segment loss rate and the congestion control algorithm shift $\hat{G}_T$ towards lower load levels, which exhibit lower loss rates. This behaviour is similar to the DVB-RCS2 normative load control, when G_T is forced to a value lower than G^*: TCP congestion control algorithm leads $\hat{G}_T$ to oscillate around a stable equilibrium point without any centralized entity that adopts complex load control strategies.

We recall that $\lambda = G/N$ is the offered load per RCST. Figure 5.16 shows that, in the presence of few RCSTs (up to 100, approximately), a reduced number of larger time-slots (as for WF 14) should be preferred, in order to allow each RCST to take as much advantage as possible of the available resources. On the other hand, when the number of RCSTs increases, a WF that offers more time-slots (as for WF 3) should be preferred in order to allow a larger population to use the channel; in fact, even if λ shows little value variations among 100 and 400 RCSTs, the aggregate throughput in Figure

Table 5.8: WF 14, MSS = 173 bytes, 64 time-slots per RA block. Relative error η when comparing the accuracy of the model presented in Section 5.2.4 and, in the last two columns, when plugging the *BLR model* in (5.20) and (5.26)

$\#RCSTs$	$NewRenoSAT_{noTO}$	$NewRenoSAT$	$BLRmodel_{noTO}$	$BLRmodel$
30	0.028	0.031	0.063	0.069
40	0.002	0.002	0.058	0.067
50	0.001	0.007	0.040	0.047
60	0.029	0.036	0.015	0.021
70	0.023	0.015	0.046	0.051

5.15 for the same load range is substantially larger or, at least, equal. It is worth noting in Figure 5.16 that λ has approximately the same value for $N > 400$, independently from WF and MSS. Finally, Figure 5.16 confirms that the scenario WF 14 and MSS = 23 bytes entails low performance and a large sub-utilization of available resources: thus, the choice of a WF and a MSS value which leads to $f > 1$ should be avoided.

5.2.7 Considerations and discussion

In this work, the performance of TCP NewReno has been analysed over a random access satellite channel. Our proposed throughput estimation model, namely NewRenoSAT, accurately fits the simulation results for the satellite scenario under consideration, where losses are only due to collisions. Furthermore, our approach has been compared with other models in literature, showing that it achieves a lower estimation error. Our simulation results support the hypothesis that collisions can be considered as random events on a RA channel. Furthermore, a simple but effective model has been provided to estimate the loss event rate p and the segment loss rate q at transport layer; this is the *BLR model* that has shown a close match with simulation results, when used in the NewRenoSAT model.

In addition, the use of TCP enforces stability in RA channels, removing the need of the DVB-RCS2 load control mechanism, which is strongly scenario-dependent. TCP acts in a distributed way, without the need of a centralized entity in charge of enforcing potentially complex load control strategies. The use of different waveforms and MSS sizes has been analysed, in order to iden-

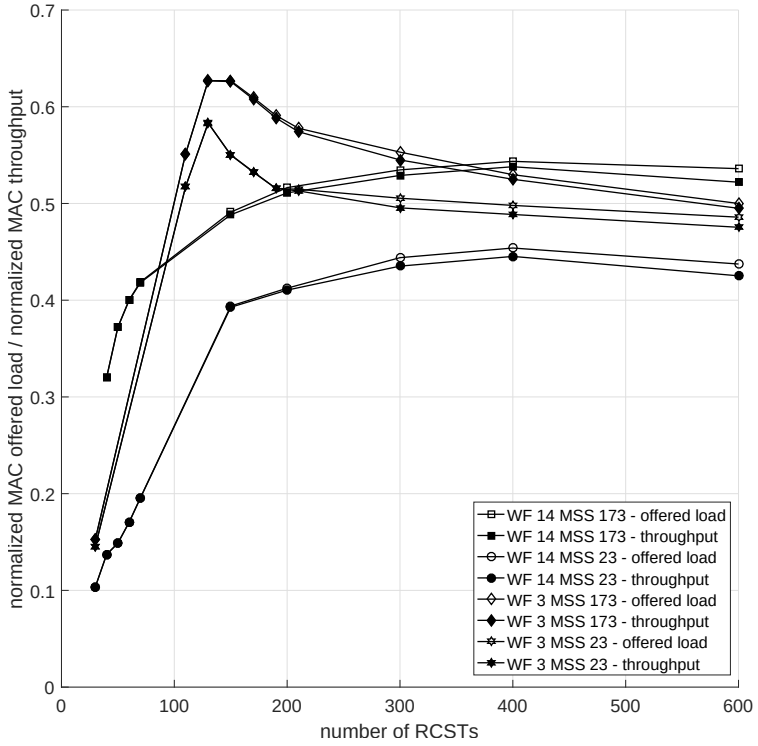


Figure 5.15: Aggregate normalized MAC throughput and aggregate offered load vs number of RCSTs

tify the combination providing the best performance in the IoT/M2M scenario under consideration, also depending on the number of RCSTs. Furthermore, any system configurations that lead to the fragmentation of TCP segments at MAC layer should be avoided because of the poor offered performance. In conclusions, the use of a TCP-based application protocol MQTT-like can support a large population in a satellite system, exploiting the advantages offered by the PUB/SUB paradigm in this context.

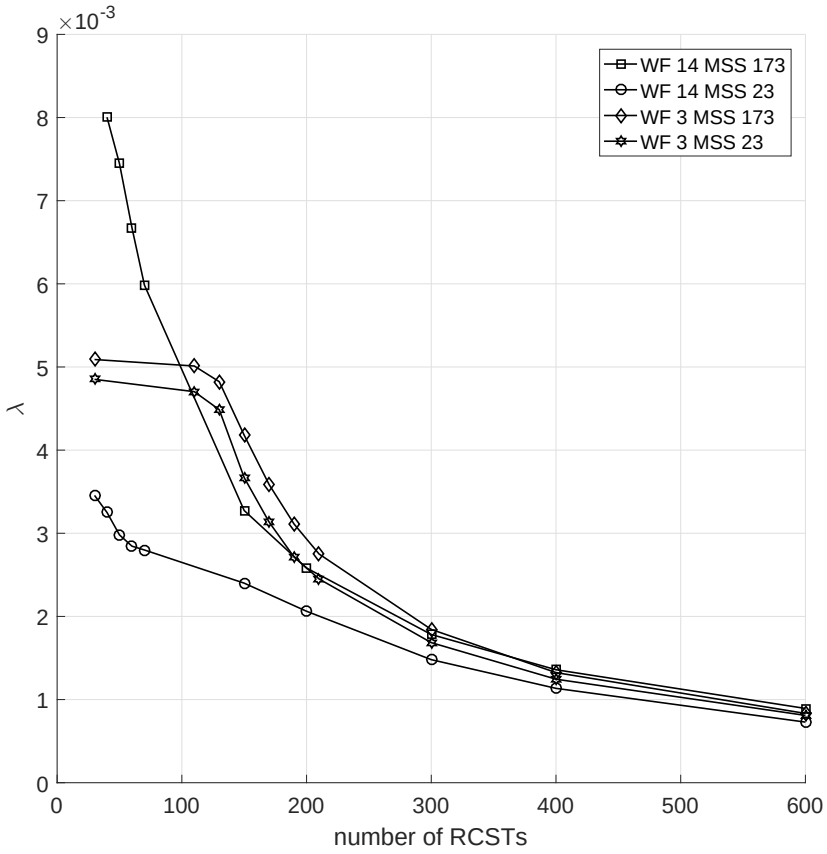


Figure 5.16: Offered load per RCST λ vs number of RCSTs

Chapter 6

RLNC in Satellite Networks: A Cooperative Scenario for Delivering M2M Traffic

Taking into account the work carried out in the NC Research Group (NWCRG) of the Internet Research Task Force (IRTF) [71], this Chapter contains the outcomes of a study carried out within the "Network Coding Applications in Satellite Communication Networks" working group of the ESA funded project Satellite Network of Excellence (SatNEx) IV.

The fraction of M2M traffic carried by satellite networks is increasing, and an efficient delivery is expected, in order to enable a large set of applications that can benefit of the advantages provided by the use of satellites. This Chapter analyses the use of Random Linear Network Coding (RLNC) techniques in Land Mobile Satellite (LMS) satellite channels to reliably deliver M2M traffic to mobile nodes in urban areas. The scenario under consideration foresees a cooperative coverage extension in LMS vehicular networks, where the use of RLNC techniques can remove the need for any fixed equipment on the ground. In the following, an overview on NC techniques is provided in Section 6.1, while related works are presented in Section 6.2. Then, the scenario under consideration is described in Section 6.3. The analytical model of the system is in Section 6.4, and the performance evaluation is carried out in Section 6.5,

where we show the impact of different mobility models in gossiping scenarios, and the advantages that the use of RLNC techniques can provide. Eventually, the conclusions are drawn in Section 6.6.

6.1 Introduction

The application of NC to communications networks is relatively recent and dates back to year 2000 [72]. Since then, NC has shown great potentials in correcting random packet errors, erasures, and errors introduced by malicious nodes, making it a powerful tool to achieve efficient network service delivery and network reliability. In the seminal work by Ahlswede et al. [73], NC is illustrated in the butterfly network example: a data source delivers a stream of messages to two receivers; all links can deliver up to one message per unit of time. In the example, the middle link represents a bottleneck and, to overcome that, the authors propose to send a combination of the messages instead of plain packets. Two incoming messages a_i and b_i are XORed, $a_i \oplus b_i$, then the coded packet $a_i \oplus b_i$ is sent in the middle link, reaching the two receivers in a single unit of time. The two receivers can reconstruct the whole information by XORing the network coded packet and a_i (or b_i) that comes on a separate link. In this way, the throughput of the network is increased, because less packet transmissions are needed to deliver the whole information block. Linear NC (LNC) is introduced by Li et al. in [74, 75]: the idea behind LNC is to create combinations of packets by taking coefficients from a Galois Field of size $GF(2^x)$. The original messages can be retrieved, at destination, by performing Gaussian elimination on the matrix generated from the LNC coefficients of the packets. Considering k source packets, a so-called *generation* worth of k input symbols, $n \geq k$ output symbols are generated by the coding procedure. If at least k independent symbols are received at destination, then decoding is possible. RLNC has been proposed in [76, 77] in 2003: when employing RLNC, the coefficients are randomly drawn from the finite field. The upper bound on failure probability in decoding is given in [76] on the order of the inverse of the size of the finite field, suggesting that random codes are very useful in unknown or changing network topologies, such as Vehicular Ad-Hoc Networks (VANETs).

There are several tens of possible application of NC in communication networks, to achieve the most various objectives: for instance, an increase of the achievable throughput, robustness to packet losses, robustness to link failures, reduction of complexity in routing strategies, and security purposes. In [73], NC is used to increase the achievable throughput: thanks to the multiple paths connecting the source and the destinations, less transmissions of packet combinations are needed to fully transmit the original data than without using NC techniques. The scenario under consideration in [73] is a multicast scenario, but the use of NC shows advantages also in unicast scenarios, and when taking into account wireless connectivity. Wireless connections are typically more affected by packet losses than wired connections, and the use of NC can counteract losses by introducing coding along the path. Classical erasure coding helps in protecting against losses at the cost of delay, which can be large if erasure coding is applied hop by hop to protect against variable loss rates. By using NC techniques, hop-by-hop coding is available and provides lots of advantages w.r.t. the use of erasure codes: such technique is referred to as *recoding* at intermediate nodes. RLNC is particularly suited in scenarios affected by different loss rates on the links composing the whole path [76, 77]. Losses can come also from link failures. In absence of NC techniques, routing schemes are required to re-route the information in order to exclude broken links; instead, the use of NC allows a faster recovering and re-routing is not needed, because the use of NC techniques over multiple links can ensure live-path protection from non-ergodic link failures [77]. Last but not least, NC provides a reduction of the complexity when considering multicast routing scenarios. A multicast scenario without the use of NC techniques may require the construction of complex routing strategies ; instead, if using NC techniques, a linear optimization is possible, based on the use of low-complexity distributed solutions [78, 79], where NC can be seen as a superset of routing. Finally, NC can be used to secure communications [80, 81]: NC packets cannot be decoded if the attacker is not able to collect enough information, for instance if a single node is under malicious control. On the other side, an attacker can replace coded packets with malicious content and this is more difficult to detect than plain attacks.

This work considers a scenario where M2M data are sent in a multicast

fashion from a single remote source via satellite to multiple receivers. Here, we take into account two different phenomena that can affect the reliability of the transmission: packet losses because of fading and because of presence of obstacles, and the presence of intermittent links in the terrestrial segment, due to the fact that the nodes move, originating a fast-changing network topology. The purpose of the VANET is to increase the probability that each node can successfully decode the data coming from the satellite, exploiting a cooperative diffusion strategy, also known as *gossiping* [82]. The multicast scenario under consideration is a typical use case in satellite scenarios, and the main aim of this work is in using NC techniques to ensure the largest possible coverage, while removing the need for any fixed ground equipment, which may needed to ensure a large coverage.

Generally speaking, IP multicasting is a key networking technique for reaching a large number of users with a single transmit operation. The most notable application of this technique is the use of satellites for distributing audio/video contents due to the inherent broadcast nature of satellites and their large area coverage. If relying on the Digital Video Broadcasting - Satellite services to Handhelds (DVB-SH) [83] standard - the satellite version of Digital Video Broadcasting - Handheld (DVB-H) [84] - several vehicular applications can also benefit of satellite networks in order to reach a wide set of customers. As far as vehicular applications are concerned, satellite transmission can be impaired by a number of factors in cities, such as the presence of buildings and other obstacles. To overcome the latter issue, the use of terrestrial gap-fillers has been proposed in [83], where they are referred to as Complementary Ground Component (CGC). The gap-fillers act as repeaters, extending the satellite coverage in areas where the satellite signal degrades too much because of the obstacles. It is very likely that, in the very near future, the upcoming Intelligent Transportation Systems (ITS) paradigm, together with a plethora of innovative services for customers, will foster using Road-Side Units (RSUs) alternatively to CGC, to allow short range communication in VANETs. In fact, RSUs may represent the ideal complement to existing communication infrastructures, in order to provide high mobility support in large networks. The paradigm Vehicle-to-everything (V2X) arises here, and IEEE 802.11p [85] is the de facto standard for terrestrial wireless communications. Wireless

Access in Vehicular Environments (WAVE) is an approved amendment to the IEEE 802.11 standard to support ITS applications, and its specifications support data exchanges between moving vehicles, and between vehicles and fixed equipment, like RSUs. In V2X scenarios, the satellite component may play a role as complementary network, in order to ensure the desired coverage in urban, suburban and rural areas.

6.2 Related works

Reliable multicast via satellite has been thoroughly analysed in the literature in the last years. A valuable survey is provided in [86], underlining how satellites offer a natural way to reach a large number of users, bypassing the issues that terrestrial networks may pose, such as the traversing of several terrestrial networks, which can be potentially congested; on the other hand, the use of satellites, usually GEO ones, can pose other issues: for instance, scalability, when several thousands of receivers produced by different manufacturers are contemporarily served by a GEO satellite in a multicast scenario [86]. Reliability can be an issue, too, in presence of a vast population of terminals listening to multicast transmissions, because different groups can experience different loss rates and, in absence of a feedback channel, it is impossible for the sender to determine if data has been correctly received by all the receivers. In the latter case, FEC techniques, applied at physical layer or as packet level FEC, are typically used to reduce the error rate.

In [87], the use of NC techniques in satellite scenarios is analysed in presence of GEO and low earth orbit (LEO) satellites: the scenario under consideration takes into account the presence of terrestrial repeaters, focusing on the next generations of satellite networks, which offer more features than just regular bent-pipe systems. NC support within On-Board Processing (OBP) is a promising technique, as alternative to FEC mechanisms [87]. A similar multicast scenario is described in [88], in presence of fixed and mobile terminals, where NC is proposed as part of the encapsulation protocol of the satellite stack, in order to provide transparency towards the upper layer protocols. The proposed communication architecture exploits a feedback channel in order to guarantee a reliable data transfer; anyway, the so-called feedback

implosion can be a major issue [89] to be dealt with in presence of feedback channels in multicast scenarios.

In [90], the case of multiple sources transmitting data via a satellite link to a single receiver is analysed. The sources exchanges packets prior to transmission, and these packets are coded together by relying on RLNC techniques. The different sources are spaced apart, thus introducing spatial diversity when transmitting in the satellite channel. The different geographical positions helps in reducing the system outage probability, even in deep fading conditions, as produced by the randomness of the surrounding environment. The work in [90] shows that the use of RLNC techniques can be an effective strategy to counteract random losses in communication channels, but at the expense of the channel capacity: a trade-off is necessary, which should take into account bandwidth requirements and channel statistics. Furthermore, the assumption that different sources are sufficiently spaced apart to experience different fading conditions, while being inter-connected via terrestrial links, is an unlikely scenario in real installations.

The work in [91] considers a multicast scenario in urban areas. The mobile nodes, moving according to the Manhattan Grid (MG) mobility model [92], have a DVB-SH radio interface to receive data from satellite, and they cooperate in ensuring the possible largest coverage on the ground, exploiting IEEE 802.11p-based communications thanks to a vehicular radio interface. LMS communications are then in place, according to the channel models in [93, 94]. The main objective in [91] is to limit the number of needed gap fillers, while still ensuring reliable data transfers, thanks to the mobile nodes acting as mobile gap-fillers (MGFs). This scenario is then deepened in [95], which takes into account the use of RLNC for cooperative coverage extension in LMS vehicular networks: an analytical assessment of the advantages that gossiping grants is provided, in presence of CGC. A minimum number of gap-fillers is necessary, according to the authors, in order to maintain a low outage probability and to extend the coverage. In [95], NC is applied to Multi-Protocol Encapsulation - Inter-burst Forward Error Correction (MPE-IFEC) [96] symbols by relying on the network-coded cooperative coverage extension (NCCCE) scheme in [97]. The outcomes of [95] show that there is a trade-off between the coverage and the rate at which the information can be injected

into the network, and that the larger is the number of mobile nodes, then the larger is the gain provided by such a cooperative approach.

6.2.1 Main contributions

This work considers a scenario similar to that proposed in [91, 95], aiming at limiting the number of CGC/RSU equipment on the ground, thanks to a RLNC-based cooperative approach. In this Chapter, we show that CGC/RSU equipment can be completely eliminated in urban areas, thanks to the specific characteristics of mobility in cities and to the use of RLNC techniques. The main contributions can be summarized as follows:

- we developed a custom simulator to assess the benefits that NC provides in presence of reliable multicast transmissions via satellite towards a VANET in an urban area;
- we show how the need for any fixed ground equipment can be completely eliminated, while ensuring a large coverage probability on the ground;
- we propose an analytical framework to analyze the joint effect of the vehicular cooperation and the application of RLNC techniques, with a focus on realistic mobility models and their impact on the numerical results.

6.3 System model

The scenario under consideration is shown in Figure 6.1: a remote data source multicasts M2M data via satellite towards vehicular nodes in an urban area. On the ground, the presence of RSUs allows to extend the coverage in areas where the link quality is poor because of the presence of obstacles. The data source applies RLNC techniques to outgoing data packets, and coded packets are received by vehicular On-Board Units (OBUs) and RSUs, both equipped with two radio interfaces: DVB-SH and IEEE 802.11p. The latter is used to *gossip* the received M2M data to close neighbors.

The main objective of this work is in understanding if RSUs/CGC are really needed to ensure full coverage or if the use of RLNC techniques may

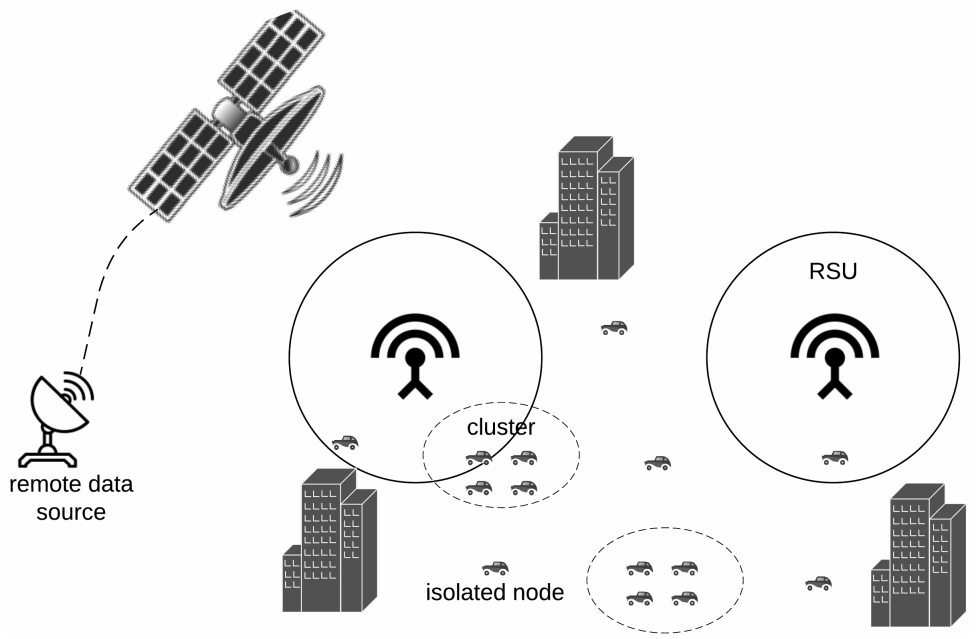


Figure 6.1: Logical description of the scenario under consideration

be sufficient to achieve, at least, the same result, at the price of a larger gossiping probability p_g, which is defined as the probability to forward (gossip) data to the neighbors within communication range. The full coverage level is here defined as the probability P_{cov} that at least $\geq 99\%$ of mobile nodes correctly decodes M2M data. We provide an analytical model in Section 6.4 that estimates the probability of correctly decoding data, when RLNC is in use, thus removing the need for RSUs. In order to further clarify the differences between CGC and RSUs in use in this work, we provide a brief comparison in Table 6.1. RSUs have the same connectivity features of a mobile node and operate as fixed equipment on the ground. In the satellite domain, CGC is defined at physical level as a signal repeater: it complements the data reception in areas where the satellite coverage cannot be guaranteed. RSUs absolve to the same function, but the hardware and the architecture is general purpose: they operate as gateways between broadband networks and local terrestrial vehicular communications, by broadcasting messages received from the satellite to 802.11p radio adapters. Both RSUs and CGC are transparent to the

CGC	**RSUs**
transparent to OBUs	transparent to OBUs
large covered area (≈ 10 [km^2])	reduced covered area (< 1 [km^2])
DVB-SH radio	DVB-SH and 802.11p radio interface
few different stakeholders	large number of potential stakeholders

Table 6.1: A brief comparison between CGC and RSUs

receivers. In a more general scenario, some vehicles may be not equipped with the satellite receiver, but they can still be acting as MGFs, because they receive data from fixed or mobile nodes within communication range. The received data is gossiped, in order to extend the coverage on the ground. In our scenario, gossiping is limited to a single hop, in order to prevent uncontrolled flooding.

In the following, we discuss the impact of the mobility model on the effectiveness of a gossip-based protocol in Section 6.3.1, the assumptions related to the satellite channel in Section 6.3.2 and the assumptions related to the terrestrial link in Section 6.3.3.

6.3.1 Impact of the mobility models

The mobility pattern in cities has specific features that affect the effectiveness of a gossiping algorithm. In cities, *clusters* of vehicles are common: for instance, in presence of traffic jams or close to traffic lights. A mobility model based on real collected traces and able to capture, at the same time, sparse and clustered network partitions, can help in simulating a more realistic scenario than relying on typical mobility models, such as MG or a Random Walk (RW) models, which are commonly used in literature. In [98], the authors propose the *Heterogeneous Random Walk* (HRW) mobility model that captures the presence of clusters as well as the isolated nodes, and the correlation between the speed of the vehicles and the clustering factor. Figure 6.2 depicts the difference among a sparse and a clustered network: the latter is typical in urban areas. On average, inside a cluster, the nodes move slower than outside it, because it is an area with a large nodes' density; instead, outside a clus-

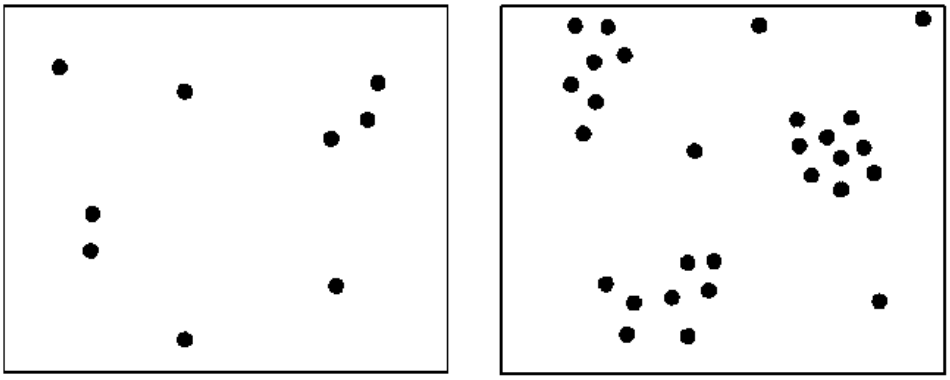


Figure 6.2: A sparse network on the left and a clustered network on the right

ter, the nodes' density decreases and, therefore, the average speed increases. The latter phenomenon has been highlighted in real mobility traces. Thus, the heterogeneous speed leads to the heterogeneous nodes' density, which is inversely proportional to the speed in [98]. The low speed and the closeness among nodes inside a cluster facilitates the data exchange. Therefore, the presence of clusters, commonly found in urban areas, improves the effectiveness of cooperative algorithms.

A VANET can be modelled as an non-empty undirected graph G: in fact, we assume that if nodes a and b are within communication range, then b receives data from a and vice versa, thus the overlay graph is undirected. The graph $G = (V, E)$ consists of a set of vertices V and a set of edges E. The neighborhood of a vertex v_i can be defined as $N_i = \{v_j : e_{ij} \in E\}$, where e_{ij} is the redirection edge connecting the vertexes v_i and v_j. Generally speaking, a vertex v_i of a digraph has indegree δ^i_{in} and outdegree δ^i_{out}, where $\delta^i_{out} \neq \delta^i_{in}$; if an undirected graph G is considered, then $\delta^i = \delta^i_{in} = \delta^i_{out}$. The average degree of a graph G is defined as $< d >= 2E/|V|$, where $|V|$ is the cardinality of V. As far as different mobility models are to be considered for the purpose of performance comparison, the first assessment should characterize the different degree of the nodes when using HRW in place of MG and RW models. HRW model exhibits a larger $< d >$ than MG and RW mobility models in the scenario under consideration, as proved in Section 6.5. It means that the average number of neighbors, whose definition is identical to $< d >$ for a

non-empty undirected graph G [99], is larger in HRW than in MG and in RW mobility models. Thus, the use of a cluster-based mobility model, such as HRW, provides several advantages: for instance, it is more realistic than MG and RW mobility models and, furthermore, it increases the probability of a correct decoding at receivers, thanks to the larger $< d >$ value.

When considering the use of HRW in our scenario, four different cases are possible: (*i*) a set of nodes originates a cluster that is inside the coverage area of a RSU; (*ii*) a set of nodes originates a cluster that is not reached by a RSU; (*iii*) a node is not inside a cluster, but is in the coverage area of a RSU; (*iv*) finally, a node is not part of a cluster and not covered by a RSU. Figure 6.1 shows the four cases so far discussed. The most favorable situation is the one described in case (*i*): the nodes can take advantage of both situations, i.e., to be inside a cluster, which is equivalent to have a lot of slowly moving neighbors, and close to a RSU. Instead, case (*iv*) represents the worst situation: a node is isolated and can only rely on the satellite channel. If RSUs are not foreseen, the most favorable situation is case (*ii*).

6.3.2 Satellite channel

In our scenario, the satellite signal is received by mobile and fixed nodes in an urban area. Mobile nodes and RSUs are equipped with DVB-SH radio interfaces. The main satellite parameters in use are provided in Table 6.2. A LMS channel, such as the one under consideration, is affected by shadowing and fading effects: it can be impaired by obstacles in the propagation path (buildings, bridges, trees, etc.), which can cause a strong attenuation. Furthermore, multipath fading occurs because the antenna of mobile users cannot be highly directive; thus, the satellite signal is received not only via the direct path, but also after being reflected by the objects in the surroundings. Due to different propagation distances, multipath signals may add destructively, thus resulting in a deep fade. In [93] and in [100], the LMS channel model in [101] is reduced to a two-state Discrete Time Markov Chain model (DTMC). The latter assumes a lossless good state (G) in Line of Sight (LoS)los (or, equivalently, a residual uncorrelated error probability), whereas the shadowing and blocking states are grouped together in a bad state (B) in NLoS (Non Line of Sight), which causes erroneous packet receptions (or, equivalently, a ne-

Parameter	Value/Description
Satellite carrier frequency	2.2 [GHz]
Satellite SNIR (in LoS)	14 dB
MODCOD	QPSK 1/5
Roll-off factor	0.35
Bandwidth	5 [MHz]

Table 6.2: System settings and parameters of the DVB-SH satellite channel

gligible success probability). The propagation conditions can be classified in urban, suburban and rural conditions. In urban and suburban environments, the main cause of channel impairment is the long-lasting shadowing of the buildings that causes intermittent satellite connectivity; in rural conditions, the main cause of impairment is the tree shadowing.

DVB-SH standard supports the use of MPE-IFEC techniques [96], an enhanced FEC module combined with two-stage interleaving layers. In our scenario, the MPE-IFEC module foresees the use of a Raptor code [102] with rate $(N, K) = (255, 170)$ and a symbol length of 1 [KB]. The ON-OFF LMS channel model derived from [93] and from the DVB-SH link layer specifications leads in what follows to an average stationary MPE-IFEC symbol loss rate $p_l \approx 36\%$ and a correlation factor $\rho \approx 0.996$. The DTMC process that represents the channel behavior is characterized by the following transition matrix T:

$$T = \begin{pmatrix} B & 1-B \\ 1-G & G \end{pmatrix} = \begin{pmatrix} 1-p_l\,(1-\rho) & p_l\,(1-\rho) \\ (1-\rho)\,(1-p_l) & p_l+\rho\,(1-p_l) \end{pmatrix} \tag{6.1}$$

In the following, we assume that each MPE-IFEC block carries exactly a RLNC symbol. In fact, before satellite broadcasting, RLNC techniques are applied, with a code rate $R_1 = k/n_1$; thus, n_1 RLNC symbols are carried by the same number of MPE-IFEC blocks. Thus, the loss probability of an MPE-IFEC block, here referred to as $BLER_{SAT}$, corresponds to the loss probability of an RLNC symbol in the DVB-SH channel. Given the two-state LMS channel model in (6.1) and a MPE-IFEC code (N, K), it is possible to derive $BLER_{SAT}$ from p_l with an approach similar to the one proposed by

Eq. (7) in [103]. The following equation provides the analytical expression of the Block Error Rate (BLER) in the satellite channel:

$$BLER_{SAT} = \sum_{j=N-K+1}^{N} P_n(j)\frac{j}{N}, \tag{6.2}$$

where $P_n(j)$ is defined as follows:

$$P_n(j) = \begin{cases} (1-p_l)(1-B)^{N-1} & j = 0 \\ p_l\big(P_n^{BB}(j) + P_n^{BG}(j)\big) + (1-p_l)\big(P_n^{GB}(j) + P_n^{GG}(j)\big) & 1 \leq j < N \\ (1-p_l)(1-G)^{N-1} & j = N \end{cases} \tag{6.3}$$

and $P_n^{BB}(j)$, $P_n^{BG}(j)$, $P_n^{GB}(j)$, and $P_n^{GG}(j)$ are defined as follows:

$$P_n^{BB}(j) = \sum_{i=2}^{\min(j,N-j+1)} B^{i-1}G^{i-1}\tbinom{j-1}{i-1}(1-G)^{j-i}(1-B)^{N-j-i+1}\tbinom{N-j-1}{i-2}$$

$$P_n^{GG}(j) = \sum_{i=2}^{\min(j+1,N-j)} B^{i-1}G^{i-1}\tbinom{j-1}{i-2}(1-G)^{j-i+1}(1-B)^{N-j-i}\tbinom{N-j-1}{i-1}$$

$$P_n^{BG}(j) = \sum_{i=1}^{\min(j,N-j)} B^{i-1}G^{i}\tbinom{j-1}{i-1}(1-G)^{j-i}(1-B)^{N-j-i}\tbinom{N-j-1}{i-1}$$

$$P_n^{GB}(j) = \sum_{i=1}^{\min(j,N-j)} B^{i}G^{i-1}\tbinom{j-1}{i-1}(1-G)^{j-i}(1-B)^{N-j-i}\tbinom{N-j-1}{i-1}.$$

Thus, each node on the ground receives $r \in [0, n_1]$ RLNC symbols after the MPE-IFEC decoding. In the following Section, we provide the vehicular channel model and discuss how the received MPE-IFEC blocks are forwarded in the vehicular channel thanks to the gossiping algorithm in use.

6.3.3 IEEE 802.11p channel

The vehicular terminals under consideration in this work, equipped with OBUs, are not energy constrained, and capable of as much computation power and memory as needed in this scenario. The link between any couple of nodes is supposed independent from the others; each node is set in promiscuous

mode for receiving any transmission in its communication range. Each data transmission is affected by the loss probability p^v_{loss} due to shadowing and fading effects, and by the collision probability p^v_{coll}. $BLER_{VEH}$ is defined as the loss probability of a RLNC symbol in the 802.11p channel, as in the following equation:

$$BLER_{VEH} = p^v_{loss} + p^v_{coll} - p^v_{loss}\, p^v_{coll}. \tag{6.4}$$

In order to trigger gossiping with probability p_g, at least k RLNC symbols must be received[1] from satellite, which occurs with probability $p_{r\geq k}$, as follows:

$$p_{r\geq k} = \sum_{i=k}^{n_1} \binom{n_1}{i} BLER^i_{SAT} \left(1 - BLER_{SAT}\right)^{n_1 - i} \tag{6.5}$$

Thus, each gossiping node receives $k' \in [k, n_1]$ RLNC symbols. In the following, we assume that, if $k' > k$, only the first k symbols are selected for gossiping. The total number of RLNC symbols that $< d >$ neighbors send to a mobile node is defined as:

$$h = \frac{k}{R_2}\, p_g \; < d > \; p_{r\geq k} = \frac{k}{R_2}\, \eta = n_2\, \eta \tag{6.6}$$

where $\eta = p_g \; < d > \; p_{r\geq k}$ is the actual number of gossiping neighbors.

In the following, we provide the analytical models to estimate p^v_{loss} in Section 6.3.3.1 and p^v_{coll} in Section 6.3.3.2. The former is based on PLA (Physical Layer Abstraction) approach [104], and the latter is based on the analytical approach in [105].

6.3.3.1 Physical Layer Abstraction - RBIR approach

In [106], a measurement campaign is provided, based on the use of Dedicated

[1] In order to correctly decode a coded transmission, at least k RLNC independent linear combinations must be received. Here, we are not assuming independence among the k symbols, thus receiving k RLNC symbols is not equivalent to say that a generation can be correctly decoded. Anyway, it is worth noting that, in presence of a large enough GF (of size $q \geq 2^8$), the probability that k random combinations are linearly independent is close to one [79].

Short Range Communication (DSRC)/IEEE 802.11p prototype radio interfaces at 5.9 [GHz]. The main contribution in [106] is a dual slope model for the path loss in urban V2X scenarios, based on real measurements. Here, the models in [106, 107] are adopted for the path loss simulator. An Orthogonal Frequency-Division Multiplexing (OFDM) signal with 52 carriers (48 information carriers) and a convolutional encoder with rate 1/2 are in use. We assume a finite fixed communication range. PLA has been implemented to simulate the vehicular channel, as proposed in the Received Bit Information Rate (RBIR) approach. The purpose of PLA is to predict the corresponding BLER, here denoted as p_{loss}^{v}, according to the instantaneous channel state information [104]. The RBIR model, here implemented, measures the mutual information at the input of the soft demapper at the receiver side. The computation of the mutual information per coded bit can be derived from the received symbol-level mutual information: for a Single Input Single Output (SISO)/Single Input Multiple Output (SIMO) system the Symbol mutual Information (SI) is given in the following formula:

$$\begin{aligned} SI(SNIR_s, m(s)) &= \log_2 M - \frac{1}{M}\sum_{m=1}^{M} E_U = \\ &= \left\{ \log_2 \left[1 + \sum_{k=1, k\neq m}^{M} exp\left(- \frac{|X_k - X_m + U|^2 - |U|^2}{1/SNIR_s} \right) \right] \right\}, \end{aligned} \tag{6.7}$$

where U is a zero mean complex Gaussian random variable with variance $1/(2\,SNIR_s)$ per component, M is the size of the modulation constellation, $SNIR_s$ is the post-equalizer SNIR at the s-th symbol or sub-carrier and $m(s)$ is the number of bits at the s-th symbol (or sub-carrier). Assuming that C sub-carriers are used to transmit a coded block, the normalized RBIR is given in the following formula:

$$RBIR = \frac{\sum_{s=1}^{C} SI(SNIR_s, m(s))}{\sum_{s=1}^{C} m(s)}. \tag{6.8}$$

We define p_{loss}^{v} as follows:

$$p_{loss}^{v} = \frac{1}{2} erfc[(RBIR - \alpha_1)/\alpha_2], \tag{6.9}$$

Parameter	Value
V2X carrier frequency	5.9 [GHz]
V2X transmission power	16 dBm
Number of 802.11p OFDM carriers	52
802.11p subcarriers spacing	0.15625 [MHz]
MCS	QPSK 1/2

Table 6.3: Settings of the 802.11p-based vehicular channel

where α_1 and α_2 are two parameters which provides a close fit to the Additive White Gaussian Noise (AWGN) performance curve, and $erfc(\cdot) = 1 - erf(\cdot)$ is the *complementary error function*. The aforementioned p^v_{loss} formulation is similar to the one in [108]. When using QPSK 1/2, for instance, $\alpha_1 = 0.5512$ and $\alpha_2 = 0.0434164$. The system parameters in use for PLA implementation in this work are readable in Table 6.3. According to those values and to RBIR expression in (6.8), p^v_{loss} can be read in Figure 6.3 as function of the distance d [m], when assuming a noise floor of -90.4 dBm.

6.3.3.2 Collision probability

The medium access mechanism of IEEE 802.11 standard relies on a carrier sense multiple access with collision avoidance (CSMA/CA) algorithm to mediate the access to the shared medium. Briefly, a backoff time (measured in slot) is chosen randomly in the interval $[0, W)$, where W is the contention window, whose range is $[W_{min}, W_{max}]$. W is doubled after each unsuccessful transmission, up to the maximum value equal to $(W_{max} + 1)$. In [105], a two-dimensional Markov chain of $(b+1)$ backoff stages is used to model the backoff time of a node. In [105], the model assumes that each packet collides in a slot time with constant and independent probability p^v_{coll}, i.e., at least one of the η neighbors contending the channel transmits in the same slot time. Therefore, if each of the η nodes transmits a packet, with channel access probability π in steady state, p^v_{coll} can be written as follows:

$$p^v_{coll} = 1 - (1 - \pi)^{\eta}. \tag{6.10}$$

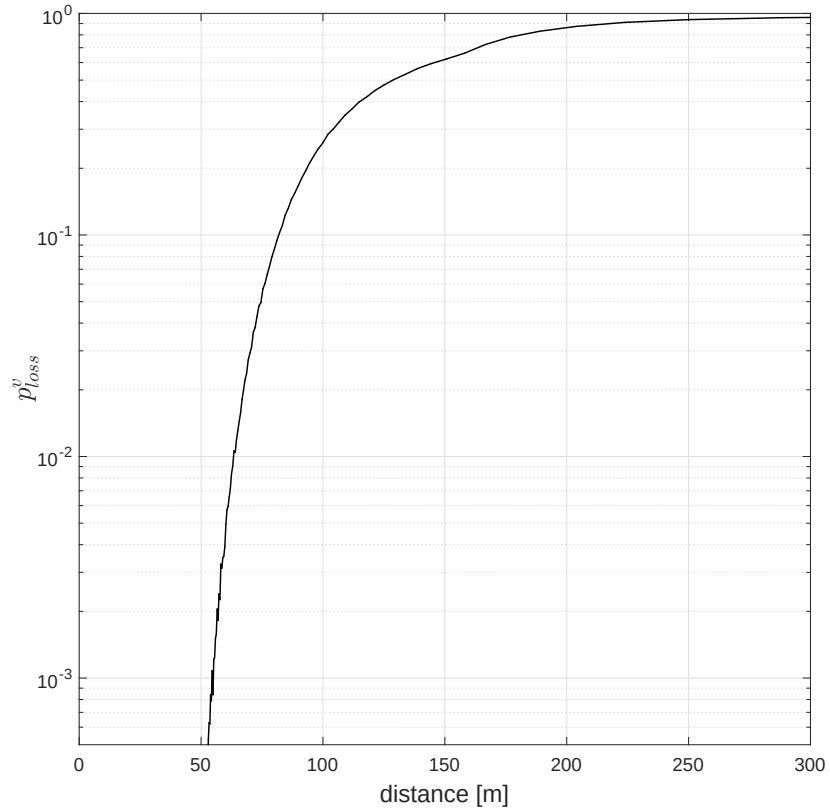


Figure 6.3: p^v_{loss} versus the distance between two nodes in the vehicular segment: at a distance $d = 150$ [m], $p^v_{loss} \approx 0.07$.

π is derived in [105] as a function of the number of backoff stages b, of the minimum contention window value W_{min}, and of the collision probability p^v_{coll} as follows:

$$\pi = \frac{2}{1 + W_{min} + p^v_{coll} W_{min} \sum_{j=0}^{b-1} (2p^v_{coll})^j}. \tag{6.11}$$

A system of non linear equations can be obtained from (6.10) and (6.11), which allows a unique solution (p^v_{coll}, π). Then, the saturation throughput can be computed. We assume to operate under saturation conditions in our

Parameter	Value
W_{min}	15 slots
W_{max}	1023 slots
b	$\log_2 \frac{W_{max}+1}{W_{min}+1}$
slot time	13 [μs]

Table 6.4: 802.11p simulator settings

scenario. 802.11p parameters have been set in the simulator as in Table 6.4, accordingly to [109].

6.4 Coverage probability: an analytical model

In this Section, we provide the analytical formulation of the coverage probability, or full coverage, as previously defined in Section 6.3. We recall that a code rate R_1 is applied to a RLNC generation before satellite broadcasting: thus, RLNC techniques at the remote source are applied equivalently to a FEC code, just above the MPE-IFEC scheme, hence increasing the probability of a correct decoding at destination. In the vehicular segment, RLNC recoding with rate R_2 is done before gossiping with probability p_g, if at least k RLNC symbols have been received. It is worth noting that the satellite and the vehicular channels typically show different loss rates, which opens to the possibility of the use of the RLNC recoding feature in VANET, in order to adapt to different channel conditions. The reliable delivery of data relies on two mechanisms: a proactive coding, based on end-to-end RLNC techniques, and a gossiping strategy in VANET.

In order to correctly decode, at least k independent RLNC symbols must be received. The cumulative distribution function of the probability of receiving k independent symbols is provided in [110] as follows:

$$P_{ns}(h,k) = \prod_{j=0}^{k-1} \left(1 - \frac{1}{q^{h-j}}\right), \tag{6.12}$$

where $q = 2^x$ is the Galois field size. Therefore, the RLNC decoding probabi-

lity after gossiping can be derived as follows:

$$P_{dec}^{RLNC}(h,k,BLER_{VEH}) = \sum_{j=k}^{h} \binom{h}{j} BLER_{VEH}^{j}(1-BLER_{VEH})^{h-j} P_{ns}(j,k). \tag{6.13}$$

Analogously, the decoding probability of the RLNC symbols received from the satellite is given by $P_{dec}^{RLNC}(n_1,k,BLER_{SAT})$. Therefore, a node can decode a RLNC generation worth of k symbols if: (i) the decoding of the symbols received via satellite is possible, according to $P_{dec}^{RLNC}(n_1,k,BLER_{SAT})$; (ii) if the decoding of the symbols received from the gossiping neighbors is possible, according to $P_{dec}^{RLNC}(h,k,BLER_{VEH})$. Eventually, we can define the global decoding probability, or *coverage probability*, as follows:

$$\begin{aligned} P_{cov}^{RLNC}(h,n_1,k) =& P_{dec}^{RLNC}(n_1,k,BLER_{SAT}) + P_{dec}^{RLNC}(h,k,BLER_{VEH}) + \\ & - P_{dec}^{RLNC}(n_1,k,BLER_{SAT}) P_{dec}^{RLNC}(h,k,BLER_{VEH}). \end{aligned} \tag{6.14}$$

Therefore, when RLNC techniques are in use, a *full coverage level* is equivalent to having $P_{cov} = P_{cov}^{RLNC}(h,n_1,k) \geq 0.99$.

6.5 Performance evaluation

A custom simulator has been developed to test the scenario under consideration. In the following simulation results, we consider a set of $|V| = 150$ nodes in an urban area large $A = 1$ [km^2]. The average vehicle speed is of 10 [m/s]. The LMS channel model in use in the following simulations is described in Section 6.3.2, and we assume what follows: a duration of the G state of ≈ 22 [s], and a duration of the B state of ≈ 15 [s], similarly to the empirical measurements provided in [100].

Table 6.5 shows the average degree $< d >$ of MG, RW and HRW mobility models. The MG and RW mobility models show a similar $< d >$ value: it means that, on average, the number of the neighbors of each node is approximately the same. The most outstanding difference between mobility in RW and in MG mobility model is in the way the nodes move: when using RW, each node can be in any point of the map, with no spatial constraints; instead,

Mobility model	**Average degree** $< d >$	**Normalized av. degree**
Random Walk (RW)	≈ 16	≈ 1
Manhattan Grid (MG)	≈ 17	≈ 1.06
Heterogeneous RW (HRW)	≈ 27	≈ 1.7

Table 6.5: Average degree of MM, RW, and HRW mobility models when simulating $|V| = 150$ mobile nodes in an urban area of 1 km^2.

when using MG, the nodes move accordingly to a squared road system. Even considering the aforementioned difference, these two mobility models show comparable $< d >$ values: therefore, according to the said metric, the two mobility models are almost indistinguishable. Instead, when using the HRW mobility model, the $< d >$ value is approximately 1.7 times larger: this is due to the clustering phenomenon, common in urban areas. Because the latter mobility model is built upon real traces, the numerical results in what follows are provided assuming the use of the HRW mobility model.

In the following, we provide the coverage probability P_{cov} in three different scenarios, assuming a generation size $k = 4$:

1. *RSUs* scenario: in Section 6.5.1, we estimate the needed RSU density in order to satisfy the requirement on P_{cov}. No gossiping is done in this scenario;
2. *VC + RSUs* scenario: in the scenario under consideration in Section 6.5.2, gossiping is done without recoding, and we estimate the minimum p_g that guarantees full coverage. Different RSUs densities are taken into account and the results are provided;
3. *VC + RLNC*: in Section 6.5.3, we remove RSUs from the urban area, while enabling recoding; thus, we estimate the minimum needed p_g at different R_2 rates.

Then, the three scenarios are compared in Section 6.5.4. In order to understand the impact of the generation size k on the coverage probability P_{cov}, in Section 6.5.5 we provide numerical results on the relation between k and the average number of neighbors $< d >$.

6.5.1 RSU scenario

In this scenario, we provide the P_{cov} as a function of the number of RSUs and the probability $P_{dec}^{RLNC}(n_1, k, BLER_{SAT})$; thus, no gossiping is done in the terrestrial part. We assume that the LMS channel between a RSU and the satellite is always in G state. Each RLNC block received by a RSU is forwarded to the $< d >$ neighbors in proximity and, because of the absence of cooperation, a mobile node can decode only if it receives the whole generation from the satellite or from a RSU. We also assume that RSUs are placed in a regular grid and, if the coverage radius of two or more RSUs is overlapped, *graph coloring* techniques are applied, i.e., the transmission frequency of a RSU is different from the one of another RSU. Thus, p_{coll}^{v} can be considered as negligible. If no coloring is necessary, i.e., no overlap among the coverage radius of close RSUs is present, while still having the whole area fully covered, we assume a RSUs density $\gamma = 1$: in the following, a RSUs density equal to one is equivalent to place approximately 44 RSUs over the urban area under consideration. Figure 6.4 shows P_{cov} versus RSUs density in this scenario for different R_1 rates. When $R_1 = 1/2$, full coverage can be achieved even in absence of RSUs: in fact, a lot of available bandwidth is traded in order to correct erasures. Instead, if $R_1 = 2/3$ or if $R_1 = 4/5$, the presence of RSUs is needed to ensure coverage: the larger the code rate, the larger the needed RSUs density in order to satisfy the requirement on P_{cov}.

Each RSU has a fixed cost for installation, and a large number of them may be needed to ensure full coverage. In order to remove the need for RSUs, and therefore lowering the cost of such a setup, we investigate the benefits provided by terrestrial cooperation in Section 6.5.2.

6.5.2 VC+RSUs scenario

In this scenario, the mobile nodes on the ground gossip the RLNC symbols received from the satellite. No recoding is done, which is equivalent to having $h = k\eta$ in (6.6). Figure 6.5 shows the numerical results: each plot begins at the minimum p_g that is necessary to have, on average, at least one neighbor (be it a RSU or a mobile node), in order to exploit the advantages that gossiping provides. Three different γ values, from 0.0225 to 1, have been

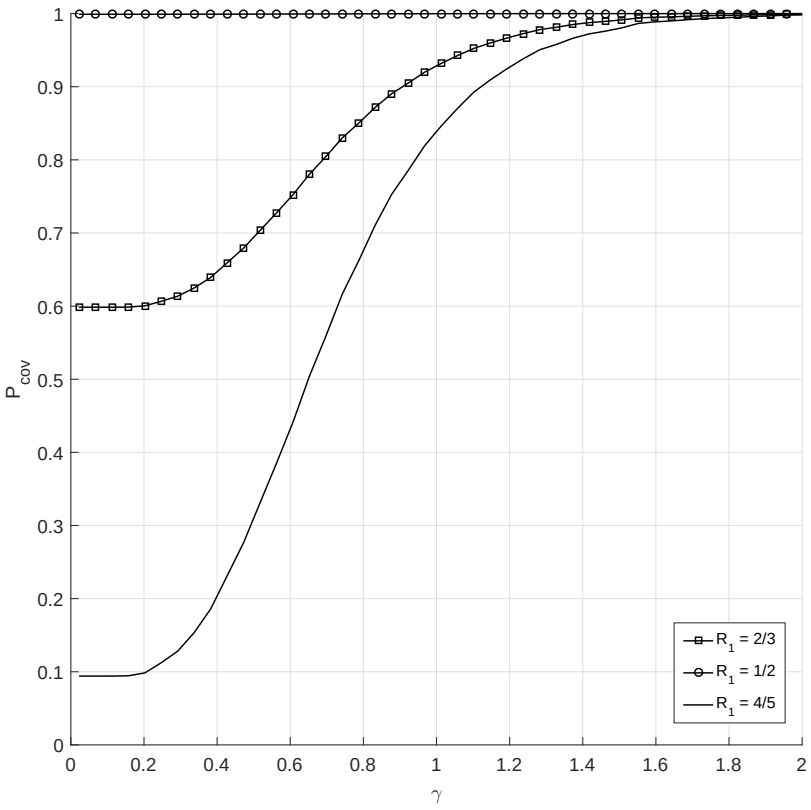


Figure 6.4: RSUs scenario: P_{cov} as a function of the RSUs density γ

taken into account, in order to show the incremental advantage provided by their presence on the ground. Furthermore, only code rates $R_1 = 2/3$ and $R_1 = 4/5$ are taken into account, because the use of a code rate $R_1 = 1/2$ does not require neither RSUs nor gossiping to guarantee the required P_{cov}, as shown in Figure 6.4. The minimum p_g ranges into $[0.4, 0.45]$ for a code rate $R_1 = 2/3$; instead, for a code rate $R_1 = 4/5$, the minimum p_g ranges into $[0.45, 0.55]$. It is worth noting that, for a fixed R_1, as p_g increases, the benefits provided by the presence of RSUs decreases: in fact, the plots converge to the

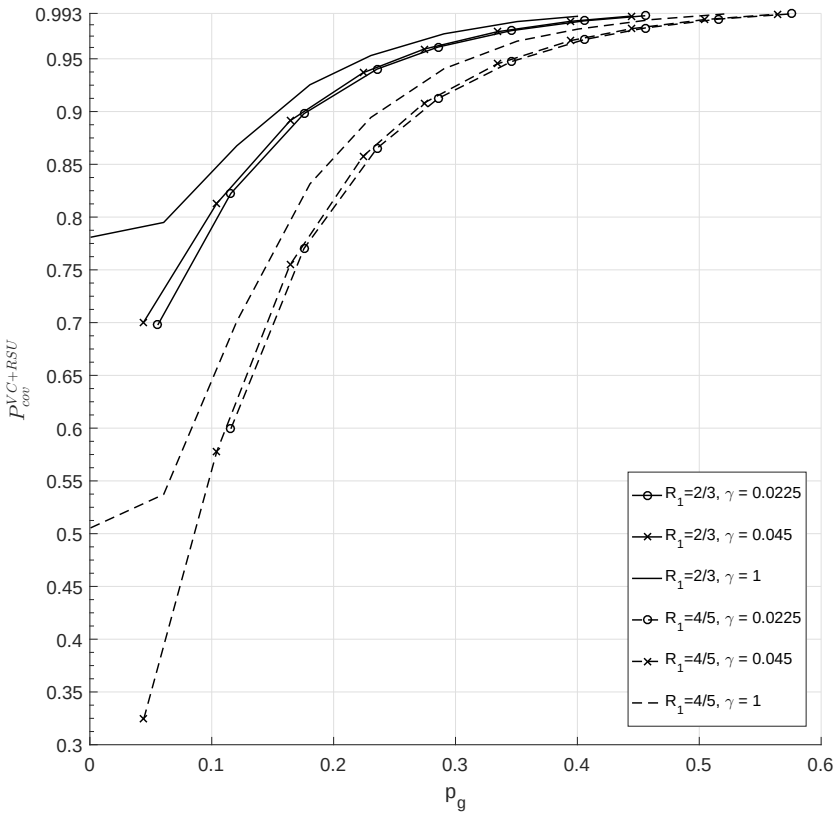


Figure 6.5: VC + RSUs scenario: P_{cov}^{VC+RSU} as a function of p_g at different γ values

same P_{cov}^{VC+RSU} value.

In the next Section, we remove RSUs and evaluate if full coverage can still be guaranteed at a price of a larger gossiping probability.

6.5.3 VC + RLNC scenario

In this scenario, recoding is done before gossiping, and RSUs are removed from the urban area. Thus, the mobile nodes rely only on the neighbors to correctly receive data, if decoding the satellite transmission has failed. The number of gossiping nodes is η, according to (6.6), and each node recodes

the received RLNC symbols with rate $R_2 = k/n_2$, where n_2 can be chosen accordingly to the channel statistics or to the target p_g. In fact, if a receiver is able to collect k independent RLNC symbols, then it is able to decode the RLNC generation: k symbols can be collected from a single neighbor (low p_g, large n_2) or from multiple neighbors (high p_g, low n_2), and p_g can be tuned accordingly to the chosen n_2 value, or vice versa. In the following, we show the numerical results for n_2 ranging into $[1, 8]$, which are the random linear combinations sent to $< d >$ neighbors from each mobile node that has collected at least k symbols. The larger the number of gossiping nodes, the larger the collision probability that can prevent the correct reception of gossiped symbols: thus, a trade-off must be identified, in order to satisfy the requirement on P_{cov}^{RLNC}. The numerical results are shown in Figure 6.6 for rates $R_1 = 2/3$ and $R_1 = 4/5$: it is visible how reducing n_2 requires a larger p_g (and vice versa) to satisfy the requirement on the target coverage probability. In Figure 6.6, the values (p_g, n_2), needed to reach the target coverage probability, range into $[(0.2, 5.5), (1, 1.5)]$ for a code rate $R_1 = 2/3$; instead, for a code rate $R_1 = 4/5$, (p_g, n_2) ranges into $[(0.2, 8), (1, 2)]$.

In order to better understand the advantages brought by the use of RLNC techniques (or, alternatively, by the presence of RSUs, as discussed in Section 6.5.2), we compare the achieved P_{cov}^{VC+RSU} with P_{cov}^{RLNC} in the same p_g range: the former is visible in Figure 6.5, the latter in Figure 6.6. At a rate $R_1 = 2/3$, for $p_g = 0.4$, a RSU density $\gamma = 1$ is needed to target the desired P_{cov}^{VC+RSU}; if $p_g = 0.45$, γ decreases to 0.0225. Instead, for the same p_g values and rate R_1, from $n_2 = 0.5$ to $n_2 = 3$ linear combinations must be sent per gossiping node in absence of RSUs, in order to satisfy the requirement on P_{cov}^{RLNC}. At a rate $R_1 = 4/5$, for $p_g = 0.45$, the minimum RSU density is $\gamma = 1$; for $p_g = 0.55$, the minimum RSU density decreases to $\gamma = 0.0225$. Instead, for the same p_g values and rate R_1, from $n_2 = 2.5$ to $n_2 = 4$ linear combinations must be sent per gossiping node, in absence of RSUs. Thus, respecting the requirement on P_{cov} is possible when removing the need for RSUs: a proper set of the n_2 and p_g values is sufficient to guarantee the desired coverage probability.

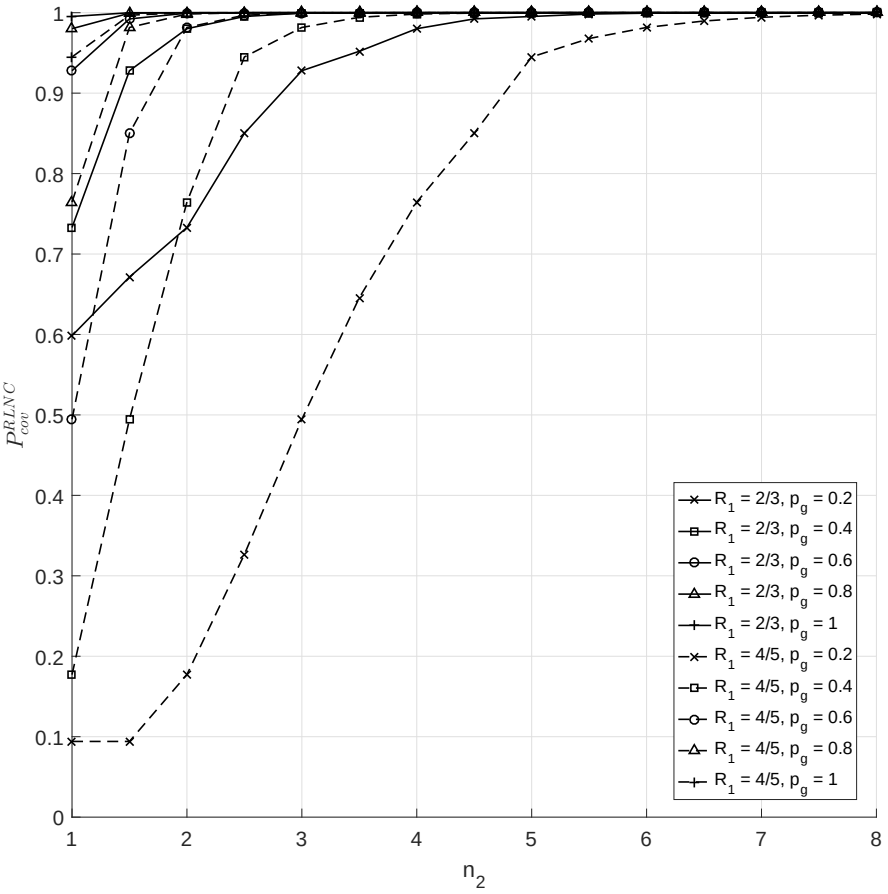


Figure 6.6: Coverage probability P_{cov}^{RLNC}, in absence of RSUs, versus the number of recoded linear combinations sent by a source node to $< d >$ neighbors

6.5.4 Comparison among the three scenarios under consideration

In this Section, we compare the three different scenarios, in order to understand the amount of circulating traffic in the terrestrial segment that can satisfy the requirement on the coverage probability. Thus, in Table 6.6, we

Scenario	**Sent packets per node** h/η	**Parameter values**	h/η	η
RSUs	k	$p_g = 0$, $\gamma = 1.7$, $R_1 = 2/3$	4	1
		$p_g = 0$, $\gamma = 2$, $R_1 = 4/5$	4	1
VC + RSUs	$k\,(1 + 1/\eta)$	$p_g = 0.4$, $\gamma = 0.02$, $R_1 = 2/3$	4.4	≈ 8
		$p_g = 0.6$, $\gamma = 0.02$, $R_1 = 4/5$	4.6	≈ 11
VC + RLNC	n_2	$p_g = 1$, $n_2 = 1.5$, $R_1 = 2/3$	1.5	≈ 16.7
		$p_g = 1$, $n_2 = 2$, $R_1 = 4/5$	2	≈ 16.7

Table 6.6: Average numerical value of the packets that each (fixed or mobile) node sends to its neighbors in order to reach the target coverage probability in the scenarios under consideration

provide the average number of symbols that each node must forward to its neighbors, in the scenarios under consideration. In the first row, the RSUs scenario is analysed: this is the scenario with the largest cost, because the coverage probability depends only on the presence of RSUs, whose installation has a fixed cost; on the other hand, it is the scenario with the lowest number of symbols forwarded by each fixed node to the mobile nodes. In the second row, the scenario VC + RSUs is analysed: here, the RSUs density is reduced and the mobile nodes cooperates: that allows to reduce the overall cost of the system at the price of a non-zero gossiping probability, which in turns means that the number of symbols that each node forwards to its neighbors must increase with respect to the RSUs scenario. In the last row, the scenario VC + RLNC is analysed: here, removing the need for RSUs provides a further cost reduction, at the price of an increasing gossiping probability p_g, i.e., larger terrestrial cooperation among the mobile nodes. It is worth noting that a larger p_g implies a larger spatial cooperation, where also groups of few isolated nodes may have a higher probability to correctly decode data; on the other hand, a higher recoding rate R_2 ensures more robustness to losses, which can prove to be useful if the vehicular channel is prone to high loss rates.

Thus, the need for RSUs can be completely removed in the scenario here under consideration, and the target coverage probability can be reached with proper settings of the gossiping probability, code rate R_1 before satellite broadcasting, and recoding rate R_2 before gossiping. The use of a realistic

mobility model, such as the HRW model here taken as reference, helps in better understanding the advantages that the use of RLNC techniques can offer in urban areas in reliable satellite multicast applications.

6.5.5 The relation between the generation size and the average number of neighbors

In this Section, we analyse the relation between k and $< d >$ and its impact on the coverage probability. We focus on the last proposed scenario, where the use of RSUs is not foreseen. We recall that, when RLNC techniques are in use, a receiver collects coded symbols from its $< d >$ neighbors, thanks to gossiping: if it succeeds in collecting at least k independent blocks of a generation, then a successfully decoding phase is possible; otherwise, decoding fails. The lower the ratio $k / < d >$, the higher the probability to correctly decode a generation, even for low p_g and n_2 values. In order to prove the latter, we provide P_{cov}^{RLNC} in Figures 6.7 and 6.8 for code rates $R_1 = 2/3$ and $R_1 = 4/5$, respectively, for an increasing $k / < d >$ ratio.

Figure 6.7 shows that a RLNC generation worth of $k \approx 1.48 < d >$ symbols can be decoded with a very high probability if $p_g = 1$ and $n_2 = 8$. We recall that, here, a coding ratio $R_1 = 2/3$ is applied to data before satellite broadcasting. If the $k / < d >$ ratio still increases, a correct decoding is unlikely to occur, as shown in the last plot of Figure 6.7. If the coding ratio under consideration is $R_1 = 4/5$, as in Figure 6.8, then a RLNC generation worth of $k \approx 0.52 < d >$ symbols can be decoded with a very high probability if $p_g = 1$ and $n_2 = 8$. Increasing the $k / < d >$ ratio beyond ≈ 0.52 may prevent a correct decoding, as shown in the last two plots of Figure 6.8.

Thus, the generation size and the coding/recoding ratios have a clear impact on the coverage probability, once the average number of neighbors is known. The average number of neighbors depends on the chosen mobility model: in this work, we choose the HRW model because it is constructed upon real mobility traces, thus providing the opportunity to better understand the impact that the use of RLNC techniques can have in urban areas. If a different mobility model is taken into account, the different average degree $< d >$ must be considered as an input for properly setting the system parameters.

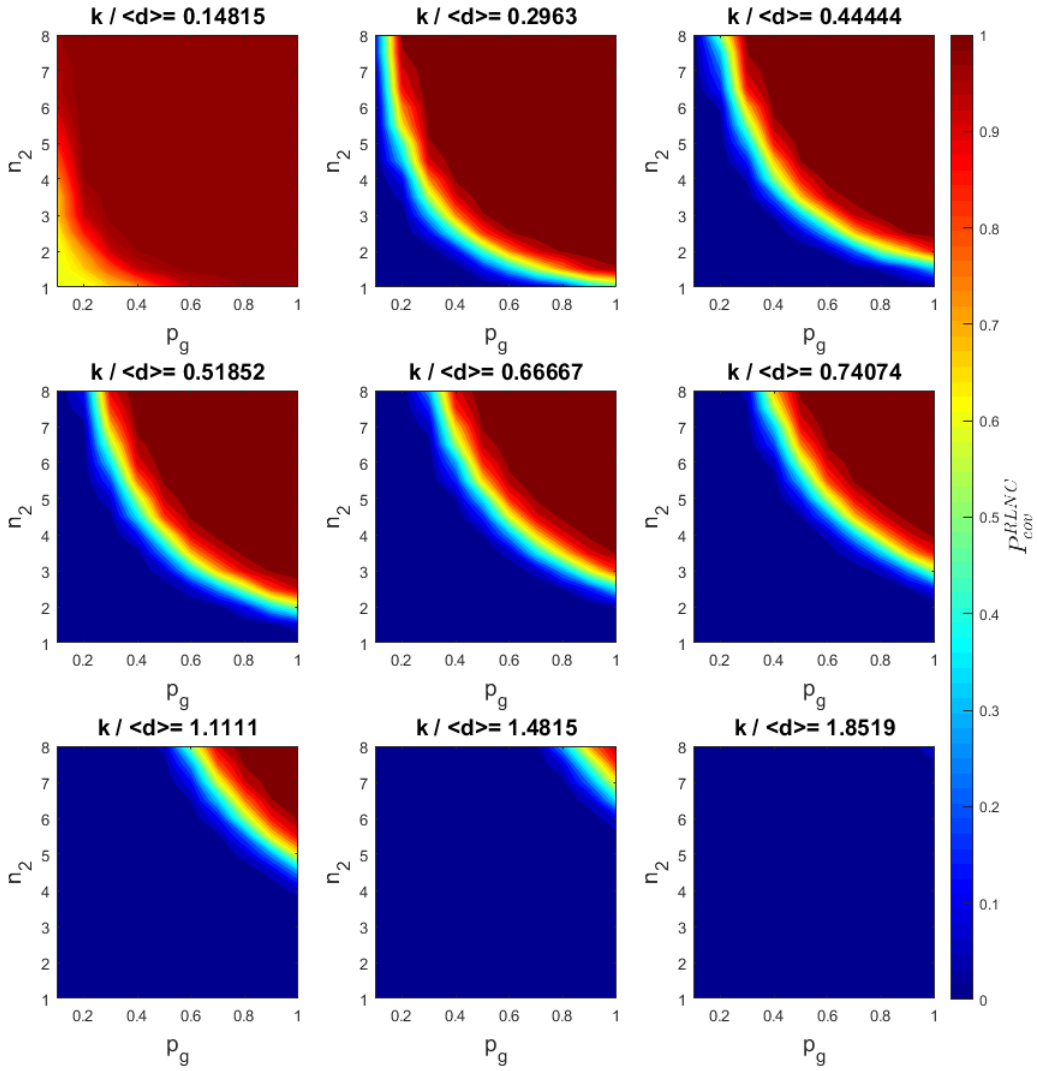


Figure 6.7: Coverage probability as a function of the generation size and the average number of neighbors for an increasing ratio $k / < d >$, when $R_1 = 2/3$

6.6 Considerations and discussion

In this Chapter, we investigated the use of RLNC techniques over LMS satellite channels to extend the satellite coverage in urban areas. Mobile nodes

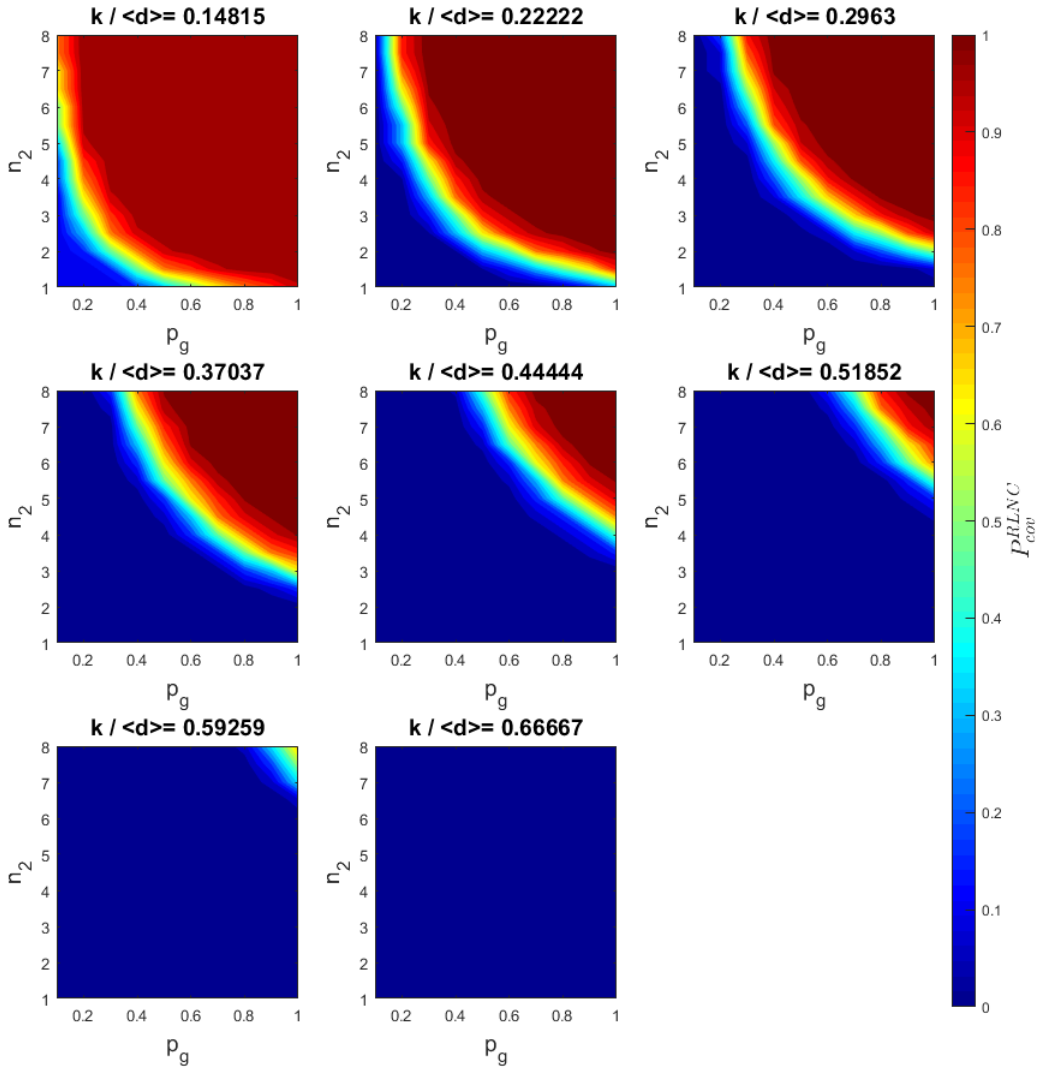


Figure 6.8: Coverage probability as a function of the generation size and the average number of neighbors for an increasing ratio $k / < d >$, when $R_1 = 4/5$

can effectively cooperate by relying on RLNC-based gossiping algorithms, removing the need for any fixed equipment on the ground meant for coverage

extension. We have provided an analytical model that takes into account the average number of neighbor nodes on the ground, in order to understand how effective the cooperation among mobile nodes can be, and the recoding feature that RLNC techniques can provide.

A mobility model built upon real traces has been used in the simulator, in order to provide numerical results as realistic as possible. The results show that RLNC-based gossiping can efficiently complement the satellite signal in ensuring a large coverage probability in urban areas, where cluster of vehicles can exchange M2M data coming from a remote source. Several parameters can be finely tuned, in order to adapt the proposed setup to different conditions, such as severely impaired communication channels, varying number of neighbors, and size of the RLNC generation.

Chapter 7

Conclusions

The pressure of IoT/M2M traffic is expected to increase in the next years. Terrestrial telecommunication systems are rapidly evolving to ensure the support to IoT/M2M services and applications provided by several market fields. In this context, the role of satellites may become crucial, and this Thesis has discussed several ideas, algorithms and techniques to improve the performance of IoT/M2M protocol stacks via RA satellite links. In order to assess the achievable performance via satellite links, analytical and empirical analyses have been provided, highlighting the most promising solutions. The following Section thoroughly recaps the achieved results, while Section 7.2 briefly proposes future research directions.

7.1 Summary of Contributions

In the context of IoT/M2M traffic via RA satellite links, this Ph.D. thesis provides contributions in two research tracks: (*i*) the maximization of MAC throughput in synchronous RA satellite links and (*ii*) the performance evaluation of the most common IoT/M2M protocol stack via RA satellite links. In addition, it examines the use of NC techniques to distribute M2M data to vehicular networks in urban areas, while removing the need for any fixed equipment on the ground.

Chapter 1 provides a brief overview on the objectives of this Thesis, along with what motivates the studies here presented. Chapter 2 provides

an overview of IoT/M2M services, applications and protocols, which will be used in the following of the Thesis. Chapter 3 provides the necessary background on the satellite access schemes in use in Chapter 4, 5 and 6.

Chapter 4 provides two contributions to the throughput maximization in RA channels: (*i*) an innovative load control algorithm that ensures a high performance level, independently from the number of connected RCSTs, and (*ii*) a novel hybrid protocol to allow the coexistence of M2M and non-M2M RCSTs. When using the empirical built load control algorithm proposed in the first part, the achieved performance level is higher than the one offered by CRDSA++ (3 replicas), used as a baseline for the performance evaluation. The load control algorithm relies on the use of linear codes prior to transmission: each RCST randomly draws the linear code $(n, k), k \geq 1$, from a code set C. The probability to extract a linear code over another depends on the weight ω_c associated with each linear code, and these weights are periodically updated by NCC. The latter is responsible for tracking the instantaneous offered load and to communicate to RCSTs the updated weights ω_c, according to a centralized algorithm that, in the load range $[0, G^*]$, pushes the system throughput close to the one achievable at G^*. This algorithm can be used as alternative to the frame length reconfiguration, a technique meant to operate a satellite system close to G^*; the latter technique can be difficult to be applied in presence of fast-varying load conditions, while the proposed load control algorithm converges to a local sub-optimal solution in few RTTs. Several parameters provide the possibility to finely tune the proposed load control algorithm. The second contribution is a novel hybrid protocol, designed to complement DA techniques to RA ones, in order to allow the coexistence of short and long data bursts in a RA satellite channel; note that DVB-RCS2 allows to design new access schemes based on the supported protocols. The proposed solution divides each superframe in a RA-reserved fraction and in a DA-reserved fraction. Further to this, it relies on a combination of centralized and local algorithms: RCSTs start the data transmission in RA operation mode, then they can switch to DA operation mode: (*a*) if the local queue of data pending for transmission is longer than a given threshold Q_{RA} [superframes] (local algorithm); (*b*) if NCC tracks a too high load insisting on the RA channel, thus it forces some RCSTs to complete the data transmission in

DA operation mode (centralized algorithm). Each RCST acting temporarily as a DA terminal can re-enable RA operation mode if its queue length is less than Q_{RA}: because of this, NCC forces in DA operation mode firstly the RCSTs acting as non-M2M devices; in other words, if a greedy RCST has a long queue of data pending for transmission (unlikely for IoT/M2M devices), it is forced to DA operation mode, thus reducing the contention, and consequently the collision rate, in the RA channel, mostly exploited by IoT/M2M terminals.

Chapter 5 provides several insights about the reliable delivery of IoT/M2M data via RA satellite links when using TCP-based application protocols. MQTT has been selected as reference protocol because it is commonly used in real scenarios. Two different metrics have been taken into account: the completion time, when short data bursts are considered, and the throughput, when the data are aggregated at the broker prior to transmission, in order to consider the scenario of a sustained offered traffic. When short data transmission are considered, the traffic pattern shows bursty features, thus the instantaneous collision rate may grow beyond an acceptable level: the system should be operated at low/medium loads, in order to achieve a reduced completion time. In the latter scenario, the phenomenon of tail losses may further increase the completion time when TCP congestion control algorithm can recover more than one loss per FR: in fact, triggering a RTO can reduce the completion time if a burst of losses occurs at the tail of the transmission. In the second scenario under consideration, i.e. sustained offered load, several original contributions are provided by this Ph.D. Thesis: (*i*) a throughput estimation model, namely *NewRenoSAT*, which accurately fits the simulation results of the IoT/M2M scenario under consideration, where losses are only due to collisions; (*ii*) the fact that collisions can be considered as random independent events on a RA channel, as the simulation results prove; (*iii*) a simple but effective model to estimate TCP loss event rate p and segment loss rate q at transport layer as a function of BLR at MAC layer; (*iv*) the stability enforced by the use of TCP in RA channels, removing the need for the DVB-RCS2 load control algorithm, which is strongly scenario-dependant; (*v*) finally, why packing several TCP segments in a single MAC burst should be avoided, because of the poor achievable performance level. The use of a

TCP-based application protocol (for instance, MQTT) can support a large population in a DVB-RCS2 satellite system, and it also requires no tuning, contrarily to the DVB-RCS2 normative load control algorithm.

Finally, Chapter 6 considers the scenario of M2M data distribution via satellite to terrestrial vehicular networks: a remote source generates data that is encapsulated into an UDP-based application protocol. RLNC techniques are in use prior to transmission, implemented as a shim layer between transport and network layer. The use of RLNC techniques is motivated by the need to ensure the larger possible coverage in urban areas, as alternative solution to the use of any fixed ground equipment. Fixed ground equipment (CGC/RSUs) may be completely substituted by the use of NC techniques, exploiting the possibility to recode data at intermediate nodes, whereas the recoding rate can be dynamically adapted to the number of vehicles in a given area: a lower recoding rate should be used in presence of a largely populated area, and a higher coding rate in poorly populated areas, in order to reduce the collision probability and to increase the probability of a correct reception of sent data.

7.2 Future Works

On the basis of the results provided by this Ph.D. Thesis, some future research works and directions are proposed in what follows.

The use of UDP-based application protocols, for instance CoAP, needs further studies when the reliability is guaranteed at the application layer; furthermore, if considering the use of the CoAP protocol in IoT/M2M satellite scenarios, the current specifications should be extended to include more sophisticated ARQ algorithms than the Stop-and-Wait mechanism now in use. Finally, the performance of the *push* strategy, as in the optional *observer pattern*, must be assessed w.r.t. the performance achievable if relying on MQTT.

Furthermore, the use of asynchronous RA protocols should be taken into account. To the best of the author's knowledge, and even if the analytical assessment provided in this Thesis is independent on the specific RA scheme, a rigorous performance evaluation of the behaviour of IoT/M2M protocol stacks is still absent, and it may prove of strong interest in real scenarios.

Bibliography

[1] P. Cerwall, P. Jonsson, R. Möller, S. Bävertoft, S. Carson, I. Godor, P. Kersch, A. Kälvemark, G. Lemne, and P. Lindberg, "Ericsson mobility report," *Ericsson*, June 2015. [Online]. Available: www.ericsson.com/res/docs/2015/ericsson-mobility-report-june-2015.pdf

[2] IDATE and UMTS, "Forum report 44," *Mobile Traffic Forecasts*, vol. 2020, 2010.

[3] J. Kim, J. Lee, J. Kim, and J. Yun, "M2M service platforms: survey, issues, and enabling technologies," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 61–76, 2014.

[4] Confederation of Indian Industry, "Machine-to-Machine: Vision 2020 Is India ready to seize a USD 4.5 trillion M2M opportunity?" 2013. [Online]. Available: http://www2.deloitte.com/content/dam/Deloitte/in/Documents/technology-media-telecommunications/in-tmt-tele-tech-2013-noexp.pdf

[5] M. De Sanctis, E. Cianca, I. Bisio, G. Araniti, and R. Prasad, "Satellite Communications Supporting Internet of Remote Things," *IEEE Internet of Things*, vol. 3, no. 1, pp. 113–123, 2016.

[6] J. Puttonen, S. Rantanen, F. Laakso, J. Kurjenniemi, K. Aho, and G. Acar, "Satellite model for Network Simulator 3," in *Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, pp. 86–91.

[7] "The Network Simulator NS-3." [Online]. Available: http://www.nsnam.org

[8] "European Space Agency - SatNEx IV (Satellite Network of Excellence) project." [Online]. Available: http://www.satnex4.org/

[9] European Space Agency, "M2M 'Maker-Space' for Satellite Communications (ARTES AT 7B.031)," 2016. [Online]. Available: https://artes.esa.int/funding/m2m-maker-space-satellite-communications-artes-7b031

[10] E. Casini, R. D. Gaudenzi, and O. Herrero, "Contention Resolution Diversity Slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1408–1419, 2007.

[11] O. del Rio Herrero and R. De Gaudenzi, "A high-performance MAC protocol for consumer broadband satellite systems," in *27th IET and AIAA International Communications Satellite Systems Conference (ICSSC)*, Edinburgh, UK, 2009.

[12] O. D. R. Herrero and R. De Gaudenzi, "High efficiency satellite multiple access scheme for machine-to-machine communications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 2961–2989, 2012.

[13] S. Scalise, C. P. Niebla, R. De Gaudenzi, O. del Rio Herrero, D. Finocchiaro, and A. Arcidiacono, "S-MIM: a novel radio interface for efficient messaging services over satellite," *Communications Magazine, IEEE*, vol. 51, no. 3, pp. 119–125, 2013.

[14] H. Burton and D. Sullivan, "Errors and error control," *Proceedings of the IEEE*, vol. 60, no. 11, pp. 1293–1301, 1972.

[15] A. Munari, G. Acar, C. Kissling, M. Berioli, and H. P. Lexow, "Multiple access in DVB-RCS2 user uplinks," *International Journal of Satellite Communications and Networking*, vol. 32, no. 5, pp. 359–376, 2014.

[16] M. Bacco, P. Cassará, and A. Gotta, "Generalized Encoding CRDSA: Maximizing Throughput in Enhanced Random Access Schemes for Satellite," *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 14, no. 5, December 2014.

[17] M. Bacco, A. Gotta, M. Luglio, C. Roseti, and F. Zampognaro, "A new RA-DA hybrid MAC approach for DVB-RCS2," in *IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2015, pp. 215–219.

[18] M. Bacco, T. De Cola, G. Giambene, and A. Gotta, "Advances on Elastic Traffic via M2M Satellite User Terminals," in *2015 International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2015, pp. 226–230.

[19] G. Giambene, M. Muhammad, D. K. Luong, M. Bacco, A. Gotta, N. Celandroni, E. K. Jaff, M. Susanto, Y. F. Hu, P. Pillai *et al.*, "Network coding

applications to high bit-rate satellite networks," in *International Conference on Wireless and Satellite Systems*. Springer, 2015, pp. 286–300.

[20] M. Bacco, P. Cassará, and A. Gotta, "Generalized Encoding CRDSA: Maximizing Throughput in Enhanced Random Access Schemes for Satellite," *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 2, December 2014.

[21] M. Bacco, A. Gotta, C. Roseti, and F. Zampognaro, "A study on TCP error recovery interaction with Random Access satellite schemes," in *Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2014 7th*. IEEE, 2014, pp. 405–410.

[22] M. Bacco, E. Ferro, and A. Gotta, "UAVs in WSNs for agricultural applications: An analysis of the two-ray radio propagation model," in *IEEE SENSORS 2014 Proceedings*. IEEE, 2014, pp. 130–133.

[23] M. Bacco, E. Ferro, and A. Gotta, "Radio propagation models for UAVs: what is missing?" in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, pp. 391–392.

[24] M. Bacco, T. De Cola, and A. Gotta, "TCP New Reno over DVB-RCS2 Random Access Links: Performance Analysis and Throughput Estimation," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[25] G. Santucci, "The Internet of Things: Between the Revolution of the Internet and the Metamorphosis of Objects," *Harald Sundmaeker, Patrick Guillemin, Peter Friess, Sylvie Woelfflé Vision and Challenges for Realising the Internet of Things. Cluster of European Research Projects on the Internet of Things (CERP-IoT)*, 2010.

[26] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[27] D. Boswarthick, O. Elloumi, and O. Hersent, *M2M communications: a systems approach*. John Wiley & Sons, 2012.

[28] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2M: From mobile to embedded internet," *Communications Magazine, IEEE*, vol. 49, no. 4, pp. 36–43, 2011.

[29] S. K. Datta and C. Bonnet, "A lightweight framework for efficient M2M device management in oneM2M architecture," in *Recent Advances in Internet of Things (RIoT), 2015 International Conference on.* IEEE, 2015, pp. 1–6.

[30] G. Mulligan, "The 6LoWPAN architecture," in *Proceedings of the 4th workshop on Embedded networked sensors.* ACM, 2007, pp. 78–82.

[31] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung, "A survey on the IETF protocol suite for the internet of things: Standards, challenges, and opportunities," *Wireless Communications, IEEE*, vol. 20, no. 6, pp. 91–98, 2013.

[32] N. Abramson, "The ALOHA System: another alternative for computer communications," in *Proceedings of the November 17-19, 1970, fall joint computer conference.* ACM, 1970, pp. 281–285.

[33] L. G. Roberts, "ALOHA packet system with and without slots and capture," *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.

[34] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs," in *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*, vol. 2. IEEE, 1996, pp. 392–396.

[35] R. De Gaudenzi, O. del Rio Herrero, G. Acar, and E. G. Barrabés, "Asynchronous Contention Resolution Diversity ALOHA: Making CRDSA truly asynchronous," *IEEE Transactions on Wireless Communications*, vol. 13, no. 11, pp. 6193–6206, 2014.

[36] G. L. Choudhury and S. S. Rappaport, "Diversity ALOHA - A Random Access Scheme for Satellite Communications," *Communications, IEEE Transactions on*, vol. 31, no. 3, pp. 450–457, 1983.

[37] E. Paolini, G. Liva, and M. Chiani, "High throughput random access via codes on graphs: Coded Slotted ALOHA," in *Communications (ICC), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1–6.

[38] G. Liva, "Graph-based analysis and optimization of Contention Resolution Diversity Slotted ALOHA," *Communications, IEEE Transactions on*, vol. 59, no. 2, pp. 477–487, 2011.

[39] W. Crowther, R. Rettberg, D. Walden, S. Ornstein, and F. Heart, "A system for broadcast communication: Reservation-ALOHA," in *Proc. 6th Hawaii Int. Conf. Syst. Sci*, 1973, pp. 596–603.

[40] L. Kleinrock and Y. Yemini, "An optimal adaptive scheme for multiple access broadcast communication," in *Proc. ICC*, vol. 78, 1978, pp. 7–2.

[41] D. Raychaudhuri, "Selective Reject ALOHA/FCFS: an advanced VSAT channel access protocol," *International Journal of Satellite Communications*, vol. 7, no. 5, pp. 435–447, 1989.

[42] J. Everett, *VSATs: very small aperture terminals*. IET, 1992, no. 28.

[43] "Second generation, D.V.B. Part2: Lower layers for satellite standard," ETSI EN 301 545-2, 2012.

[44] C. Roseti and E. Kristiansen, "TCP behavior in a DVB-RCS environment," in *24th AIAA International Communications Satellite Systems Conference (ICSSC): June*, 2006.

[45] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[46] M. Bacco, P. Cassarà, E. Ferro, and A. Gotta, "Generalized Encoding CRDSA: Maximizing Throughput in Enhanced Random Access Schemes for Satellite," in *Personal Satellite Services*. Springer, 2013, pp. 115–122.

[47] A. Zanella, "Estimating collision set size in framed Slotted Aloha wireless networks and RFID systems," *Communications Letters, IEEE*, vol. 16, no. 3, pp. 300–303, 2012.

[48] A. Meloni and M. Murroni, "Random Access in DVB-RCS2: Design and Dynamic Control for Congestion Avoidance," *IEEE Transactions on Broadcasting*, vol. 60, no. 1, pp. 16–28, 2014.

[49] U. of Southern California (USC), "The Network Simulator NS-2." [Online]. Available: http://www.isi.edu/nsnam/ns/

[50] R. De Gaudenzi and O. D. Herrero, "Advances in Random Access protocols for satellite networks," in *Satellite and Space Communications, 2009. IWSSC 2009. International Workshop on*. IEEE, 2009, pp. 331–336.

[51] N. Parvez, A. Mahanti, and C. Williamson, "An analytic throughput model for TCP NewReno," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 2, pp. 448–461, 2010.

[52] R. Dunaytsev, K. Avrachenkov, Y. Koucheryavy, and J. Harju, "An Analytical Comparison of the Slow-but-Steady and Impatient Variants of TCP New Reno," in *Wired/Wireless Internet Communications*. Springer, 2007, pp. 30–42.

[53] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation," *IEEE/ACM Transactions on Networking (ToN)*, vol. 8, no. 2, pp. 133–145, 2000.

[54] C. Pereira and A. Aguiar, "Towards efficient mobile M2M communications: survey and open challenges," *Sensors*, vol. 14, no. 10, pp. 19 582–19 608, 2014.

[55] G. Giambene, "Queuing theory and telecommunications networks and applications, 2nd edition, 2014, Springer NY."

[56] L. Guo and I. Matta, "The war between mice and elephants," in *Network Protocols, 2001. Ninth International Conference on.* IEEE, 2001, pp. 180–188.

[57] N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP latency," in *in IEEE INFOCOM*, 2000, pp. 1724–1751.

[58] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4. ACM, 1998, pp. 303–314.

[59] M. Mellia, H. Zhang, and I. Stoica, "TCP model for short lived flows," *IEEE Communications Letters*, vol. 6, no. 2, pp. 85–87, 2002.

[60] M. Collina, M. Bartolucci, A. Vanelli-Coralli, and G. E. Corazza, "Internet of Things application layer protocol analysis over error and delay prone links," in *Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2014 7th.* IEEE, 2014, pp. 398–404.

[61] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? A survey of alternatives," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 4–16, 2014.

[62] B. Sikdar, S. Kalyanaraman, and K. Vastola, "Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno, and SACK," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 959–971, Dec 2003.

[63] K. Zhou, K. Yeung, and V. Li, "Throughput modeling of TCP with slow-start and fast recovery," in *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, vol. 1, Nov 2005, p. 5 pp.

[64] Z. Chen, T. Bu, M. Ammar, and D. Towsley, "Comments on modeling TCP Reno performance: a simple model and its empirical validation," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 2, pp. 451–453, 2006.

[65] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 6582 (Proposed Standard), Internet Engineering Task Force, April 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6582.txt

[66] C. Liu and E. Modiano, "An analysis of TCP over random access satellite links," in *IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 4, March 2004, pp. 2033–2040.

[67] N. Celandroni and R. Secchi, "Suitability of DAMA and contention-based satellite access schemes for TCP traffic in mobile DVB-RCS," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1836–1845, May 2009.

[68] N. Celandroni, F. Davoli, E. Ferro, and A. Gotta, "On elastic traffic via Contention Resolution Diversity Slotted Aloha satellite access," *International Journal of Communication Systems*, 2014.

[69] C. Joo and S. Bahk, "Start-up transition behaviour of TCP NewReno," *Electronics Letters*, vol. 35, no. 21, pp. 1818–1820, Oct 1999.

[70] N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP latency," in *IEEE INFOCOM, 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3. IEEE, 2000, pp. 1742–1751.

[71] "Internet Research Task Force - Network Coding Research Group." [Online]. Available: https://irtf.org/nwcrg

[72] R. Bassoli, H. Marques, J. Rodriguez, K. W. Shum, and R. Tafazolli, "Network coding theory: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 4, pp. 1950–1978, 2013.

[73] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, 2000.

[74] S. Li and R. Yeung, "Network multicast flow via linear coding," in *Proc. Int. Symp. Operations Research and its Applications*, 1998, pp. 197–211.

[75] S.-Y. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE transactions on information theory*, vol. 49, no. 2, pp. 371–381, 2003.

[76] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proceedings of IEEE International Symposium on Information Theory*. IEEE, 2003.

[77] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *41th Annual Allerton Conference Communication, Control, and Computing*. Citeseer, 2003.

[78] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad-hoc networks using network coding," *IEEE Transactions on communications*, vol. 53, no. 11, pp. 1906–1918, 2005.

[79] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413–4430, 2006.

[80] N. Cai and R. W. Yeung, "Secure network coding," in *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*. IEEE, 2002, p. 323.

[81] L. Lima, M. Médard, and J. Barros, "Random linear network coding: A free cipher?" in *2007 IEEE International Symposium on Information Theory*. IEEE, 2007, pp. 546–550.

[82] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, "A survey of gossiping and broadcasting in communication networks," *Networks*, vol. 18, no. 4, pp. 319–349, 1988.

[83] ETSI, "102 584," *Digital Video Broadcasting (DVB); DVB-SH Implementation Guidelines*, vol. 34, pp. 116–130, 2008.

[84] ETSI, "302 304 v1. 1.1 (2004-11): Digital Video Broadcasting (DVB): Transmission System for Handheld Terminals (DVB-H)," *European Telecommunication Standard*, 2004.

[85] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for Wireless Access in Vehicular Environments," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*. IEEE, 2008, pp. 2036–2040.

[86] M. W. Koyabe and G. Fairhurst, "Reliable multicast via satellite: A comparison survey and taxonomy," *International Journal of Satellite Communications*, vol. 19, no. 1, pp. 3–28, 2001.

[87] F. Vieira, S. Shintre, and J. Barros, "How Feasible is Network Coding in Current Satellite Systems?" in *5th Advanced Satellite Multimedia Systems Conference (ASMS) and 11th Signal Processing for Space Communications Workshop (SPSC)*, Cagliari, Italy, Sep. 2010.

[88] F. Vieira and J. A. Barros, "Network Coding Multicast in Satellite Networks," in *Next Generation Internet Networks, 2009. NGI'09*. IEEE, 2009, pp. 1–6.

[89] M. P. Anastasopoulos, A. D. Panagopoulos, and P. G. Cottis, "A feedback suppression algorithm for reliable satellite multicast based on spatial–temporal prediction of the satellite channel," *International Journal of Satellite Communications and Networking*, vol. 27, no. 2, pp. 117–139, 2009.

[90] R. Alegre-Godoy and M. A. Vazquez-Castro, "Spatial Diversity with Network Coding for ON/OFF Satellite Channels," *Communications Letters, IEEE*, vol. 17, no. 8, pp. 1612–1615, 2013.

[91] G. Cocco, C. Ibars, and O. del Rio Herrero, "Cooperative satellite to land mobile gap-filler-less interactive system architecture," in *2010 5th Advanced Satellite Multimedia Systems Conference and the 11th Signal Processing for Space Communications Workshop*. IEEE, 2010, pp. 309–314.

[92] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad-hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[93] E. Lutz, D. Cygan, M. Dippold, F. Dolainsky, and W. Papke, "The Land Mobile Satellite communication channel-recording, statistics, and channel model," *Vehicular Technology, IEEE Transactions on*, vol. 40, no. 2, pp. 375–386, 1991.

[94] F. P. Fontán, M. Vázquez-Castro, C. E. Cabado, J. P. García, and E. Kubista, "Statistical modeling of the LMS channel," *Vehicular Technology, IEEE Transactions on*, vol. 50, no. 6, pp. 1549–1567, 2001.

[95] G. Cocco, N. Alagha, and C. Ibars, "Cooperative coverage extension in vehicular land mobile satellite networks," *Vehicular Technology, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2015.

[96] B. Sayadi, Y. Leprovost, S. Kerboeuf, M. L. Alberi-Morel, and L. Roullet, "MPE-IFEC: An enhanced burst error protection for DVB-SH systems," *Bell Labs Technical Journal*, vol. 14, no. 1, pp. 25–40, 2009.

[97] G. Cocco, N. Alagha, and C. Ibars, "Network-coded cooperative extension of link level FEC in DVB-SH," in *Proceedings of the 29th AIAA International Communications Satellite Systems Conference*, 2011.

[98] M. Piórkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "On clustering phenomenon in mobile partitioned networks," in *Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models*. ACM, 2008, pp. 1–8.

[99] R. Diestel, *Graph theory*. Springer Science & Business Media, 2005.

[100] S. Scalise, H. Ernst, and G. Harles, "Measurement and Modeling of the Land Mobile Satellite Channel at Ku-band," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 2, pp. 693–703, 2008.

[101] I. Rulands, H. Ernst, J. Kunisch, and G. Harles, "Feasibility study of a mobile Ku-band terminal: Final report," Technical report, ESA [Online]. Available: http://www. telecom. esa. int, Tech. Rep., 2002.

[102] A. Shokrollahi, "Raptor codes," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2551–2567, 2006.

[103] N. Celandroni and A. Gotta, "Performance analysis of systematic upper layer FEC codes and interleaving in land mobile satellite channels," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1887–1894, 2011.

[104] R. Srinivasan, J. Zhuang, L. Jalloul, R. Novak, and J. Park, "Draft IEEE 802.16m evaluation methodology document," *IEEE C802. 16m-07/080r2*, 2007.

[105] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, 2000.

[106] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 8, pp. 1501–1516, 2007.

[107] T. Islam, Y. Hu, E. Onur, B. Boltjes, and J. de Jongh, "Realistic simulation of IEEE 802.11p channel in mobile vehicle to vehicle communication," in *Microwave Techniques (COMITE), 2013l Conference on.* IEEE, 2013, pp. 156–161.

[108] R. Srinivasan, J. Zhuang, L. Jalloul, R. Novak, and J. Park, "IEEE 802.16m evaluation methodology document (EMD)," *IEEE 802.16 Broadband Wireless Access Working Group*, 2008.

[109] C. Han, M. Dianati, R. Tafazolli, and R. Kernchen, "Throughput analysis of the IEEE 802.11 p enhanced distributed channel access function in vehicular environment," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd.* IEEE, 2010, pp. 1–5.

[110] O. Trullols-Cruces, J. M. Barcelo-Ordinas, and M. Fiore, "Exact decoding probability under random linear network coding," *IEEE communications letters*, vol. 15, no. 1, pp. 67–69, 2011.