

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



**FEUP**

# **Load Balancing in Multi-Beam Satellites**

**Filipe Gonçalves Ferreira Gomes**

Master in Informatics and Computing Engineering

Supervisor: Daniel Lucani (PhD)

26<sup>th</sup> July, 2012



# **Load Balancing in Multi-Beam Satellites**

**Filipe Gonçalves Ferreira Gomes**

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Doctor Ana Cristina Ramada Paiva Pimenta, Prof. Auxiliar da FEUP

External Examiner: Doctor Paula Maria M. Moura Gomes Viana, Prof. Coordenador do Instituto Superior de Engenharia do Porto

Supervisor: Doctor Daniel Enrique Lucani Rotter, Prof. Auxiliar Convocado da FEUP

---

26<sup>th</sup> July, 2012



# Abstract

Telecommunication systems experience continually a traffic volume growth. Satellite systems pose as an attractive way of linking geographically disperse terminals. Their high upfront cost leads satellite operators to continually seek after performance improvement. The introduction of multi-beam satellites coupled with adaptive coding and modulation (ACM) allows for a significant increase in throughput for the same RF power and bandwidth. However, multi-beam satellites with conventional payloads provide an uniform resource assignment per beam, i.e., the same time servicing each beam. Thus, the optimal operating point of the systems lies in a uniform throughput demand across all beams.

Assuming that a fraction of the terminals can simultaneously receive from two or more beams and exploiting the fact that a significant signal overlap between beams exists, allows for additional flexibility for solving this resource allocation problem. In essence, multiple available routes can be used to deliver the traffic. In fact, each route is not required to send all data packets, as is usually the case for classical soft-handover mechanisms, but rather a fraction of the original data. The key is sending enough data packets through all available beams to ensure a successful delivery.

This thesis studies the multi-beam satellite technology and novelty techniques in this context, such as network coding. The main goal of this thesis is to assess the benefits of load balancing in multi-beam satellite networks, with the objective of increasing throughput and reducing the delay that the data packets experience in the system. The main focus of this work lies in taking advantage of unused resources by leveraging multiple, available routes. However, network coding provides additional dimensions and advantages. In particular, it can provide solutions based on flexible resource allocation both for static and dynamic loading differences amongst the beams. The former is associated with hotspots, i.e., highly loaded beams, which have in general higher traffic demands. The latter is related to the dynamic nature of the data traffic.

The work in this thesis required the development of a custom simulator to evaluate the performance improvements in different conditions. Through these simulations, we were able to attain significant throughput improvements with load balancing, specially when the system is facing uneven demand patterns.



# Resumo

Os sistemas de telecomunicações vivem um contínuo crescimento do volume de tráfego. Os satélites de comunicações apresentam-se como uma forma atrativa de ligar terminais dispersos geograficamente. O seu alto custo leva os operadores de satélites a procurar continuamente melhorias de desempenho. A introdução de satélites *multi-beam* (múltiplos feixes) juntamente com *Adaptive Coding and Modulation* (ACM) permite aumentos significativos na taxa de transferência para a mesma energia de rádio-frequência e largura de banda. No entanto, os satélites *multi-beam* com cargas convencionais fornecem uma atribuição uniforme de recursos por feixe, isto é, o mesmo tempo de serviço para cada feixe.

Assumindo que uma fração dos terminais pode receber simultaneamente de dois ou mais feixes e explorando o facto de que existe uma sobreposição significativa dos feixes, isto permite uma flexibilidade adicional para resolver este problema de alocação de recursos. Em essência, múltiplas rotas disponíveis podem ser utilizadas para fornecer o tráfego. De facto, cada rota não se encontra forçada a enviar todos os pacotes de dados, como é habitual no caso de mecanismos de *soft-handover*, mas antes apenas uma fração dos dados originais. A chave é enviar dados suficientes por todos os feixes disponíveis para assegurar uma entrega bem sucedida.

Esta tese estuda a tecnologia dos satélites *multi-beam* e técnicas inovadoras neste contexto, como codificação em rede. A meta principal desta tese é avaliar os benefícios do balanceamento de carga em redes de satélites *multi-beam*, com o objetivo de aumentar a taxa de transferência e reduzir o atraso que os pacotes de dados conhecem no sistema. O foco principal deste trabalho repousa em tirar partido dos recursos inusitados alavancando as múltiplas rotas disponíveis. Porém, a codificação em rede fornece dimensões e vantagens adicionais. Em particular, pode fornecer soluções baseadas na alocação flexível de recursos tanto para diferenças de carga estáticas e dinâmicas entre os feixes. A primeira está associada a *hotspots*, isto é, feixes altamente carregados, que têm uma procura de tráfego geralmente maior. A última está associada à natureza dinâmica do tráfego de dados.

O trabalho desta tese exige o desenvolvimento de um simulador personalizado para avaliar as melhorias de desempenho em diferentes condições. Através destas simulações, fomos capazes de atingir melhorias de taxa de transferência significativas com balanceamento de carga, especialmente quando o sistema enfrenta padrões de procura desnivelados.





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem . . . . .	1
1.2	Objectives . . . . .	3
1.3	Structure . . . . .	3
<b>2</b>	<b>State of the Art</b>	<b>5</b>
2.1	Introduction . . . . .	5
2.2	Multi-beam satellites . . . . .	5
2.2.1	Introduction . . . . .	5
2.2.2	Power allocation schemes and beam allocation techniques . . . . .	7
2.2.3	Challenges . . . . .	8
2.3	Multipath routing . . . . .	8
2.4	Network Coding . . . . .	9
2.4.1	Definition . . . . .	9
2.4.2	Benefits . . . . .	11
2.4.3	Applications . . . . .	12
<b>3</b>	<b>Problem Description</b>	<b>15</b>
3.1	Introduction . . . . .	15
3.2	Methodology . . . . .	15
3.2.1	Mathematical formulation of multi-beam load balancing . . . . .	15
3.2.2	Algorithms for multipath allocation . . . . .	16
3.2.3	Algorithms for allocation of resources . . . . .	18
3.3	Conclusions . . . . .	19
<b>4</b>	<b>Algorithms</b>	<b>21</b>
4.1	Introduction . . . . .	21
4.2	Static allocation . . . . .	21
4.3	Load balancing . . . . .	22
4.4	Packet discard policy . . . . .	27
4.4.1	Introduction . . . . .	27
<b>5</b>	<b>Evaluation</b>	<b>31</b>
5.1	Simulation model . . . . .	31
5.1.1	Clients . . . . .	32
5.1.2	Gateways . . . . .	32
5.2	Metrics . . . . .	33
5.2.1	Transmission time . . . . .	33

## CONTENTS

5.2.2	Number of transmitted, delayed, and discarded data packets . . . . .	33
5.2.3	Load from assigned sessions . . . . .	34
5.2.4	Unmet System Capacity (USC) . . . . .	34
5.3	Simulation tools . . . . .	35
5.3.1	MATLAB . . . . .	35
5.3.2	Custom application . . . . .	35
5.4	Load balancing . . . . .	36
5.4.1	Introduction . . . . .	36
5.4.2	Impact of load balancing on throughput . . . . .	36
5.4.3	Delay window effect on load balancing performance . . . . .	40
5.4.4	Computational complexity . . . . .	41
5.4.5	Discussion . . . . .	43
5.5	Fair Access Policies . . . . .	44
5.5.1	Introduction . . . . .	44
5.5.2	Results . . . . .	44
5.5.3	Discussion . . . . .	46
5.6	Redundancy and network coding . . . . .	46
5.6.1	Introduction . . . . .	46
5.6.2	Results . . . . .	47
5.6.3	Discussion . . . . .	48
<b>6</b>	<b>Conclusions</b>	<b>51</b>
6.1	Conclusions and future work . . . . .	51
<b>A</b>	<b>Appendix A</b>	<b>53</b>
	<b>References</b>	<b>55</b>

# List of Figures

1.1	Typical satellite network topology for satellite-based internet access service. From left to right: ground station connected to the Internet, satellite acting as intermediary and end user station. . . . .	2
2.1	Coverage area and beam layout, displaying frequency reuse. Each cell represents the coverage area of a given beam; each colour a given frequency. . . . .	6
2.2	Assumed traffic distribution in 2010. . . . .	7
2.3	Example of traditional multicast: the central nodes receive different packets from the source node. Each sink node receives the packets through different paths. . .	10
2.4	Example of network coding: the middle node receives packets A and B. It then broadcasts a packet, result of the combination of the incoming packets. The left and right nodes can recover the missing packet using the one they sent and the coded packet they receive. . . . .	11
2.5	Example of linear combination of two data packets using network coding. We use symbols in $GF(2^8)$ , data packets are of $n$ bits, $C_i$ represents the coefficients of the linear combination. The process ends in a coded packet, composed by coded data, a header and the encoding coefficients. . . . .	12
3.1	Example of load balancing: $T$ is the time available during a frame and the bars represent the amount of time it would take to each beam to deliver its data. We can exploit other beams' unused resources. . . . .	18
4.1	Diagram depicting the packet swaps performed by the algorithm in the second phase.	24
4.2	Diagram depicting the packet swaps performed by the algorithm in the second phase.	25
4.3	Diagram depicting the packet swaps performed by the algorithm in the second phase.	25
4.4	Comparison of results from the algorithm and an optimization. Lines represent the optimization. . . . .	27
4.5	Valid configurations in the stable demand region, obtained with the optimization formulation; charted transmission time of neighbour beams. . . . .	29
5.1	Beams grouped by gateway. . . . .	32
5.2	Improvement with load balancing vs. without: with load balancing performed globally . . . . .	37
5.3	Improvement with load balancing vs. without: with load balancing performed globally and real demand . . . . .	37
5.4	Improvement with load balancing vs. without: with load balancing performed within gateways . . . . .	38
5.5	Improvement with load balancing vs. without: with load balancing performed within gateways and real demand . . . . .	39

## LIST OF FIGURES

5.6	Comparison of improvements in percentage and number of beams: globally vs. within gateway. . . . .	40
5.7	Comparison of improvements in Mbps: globally vs. within gateway. . . . .	41
5.8	Improvements in the number of discarded packets, in the cases of different number of windows of delay tolerance. . . . .	41
5.9	Comparison of run time for simulations, with normalized demand. . . . .	42
5.10	Comparison of run time for simulations with real demand. . . . .	43
5.11	Comparison between the default and priority policies: traffic in beam 13 and traffic in the two groups of sessions discriminateComparison between the default and priority policies: traffic in beam 13 and traffic in the two groups of sessions discriminated. . . . .	45
5.12	Difference in throughput compared to the default policy: among heavier loaded sessions, less loaded sessions and globally. . . . .	45
5.13	Improvement in successfully transmitted traffic with DNC, with different erasure probabilities. . . . .	47
5.14	Improvement in successfully transmitted traffic with SNC, with different erasure probabilities. . . . .	48

# List of Tables

A.1 Beam Demand according to the DDSO study . . . . .	53
---	----

## LIST OF TABLES

# Abbreviations

ACM	Adaptive Coding and Modulation
DDSO	Digital Divide Satellite Offer
DNC	Dense Network Coding
ESA	European Space Agency
FAP	Fair Access Policy
FEC	Forward Error Correction
FMT	Fade Mitigation Technique
GSE	Generic Stream Encapsulation
IDE	Integrated Development Environment
IP	Internet Protocol
MODCOD	Modulation and Coding
QoE	Quality of (user) Experience
QoS	Quality of Service
RF	Radio frequency
RLNC	Random Linear Network Coding
SINR	Signal to Interference plus Noise Ratio
SNC	Systematic Network Coding
TWT	Traveling-Wave Tube
VoD	Video on Demand
WSN	Wireless Sensor Network





# Chapter 1

## Introduction

### 1.1 Problem

Communication satellites are instrumental for providing communication services to vast areas and with minimal infrastructure. After the upfront cost of developing and launching a satellite to space, not much more investments are needed, when comparing to the wired networking model, which relies on laying cables to each terminal, or even wireless infrastructures, which require base towers scattered along the serviced area.

Due to the increase in the number of users, the requirements of different kinds of content that is being distributed, e.g., video, voice, internet traffic, and the emergence of a broader spectrum of applications, there are stricter efficiency and capacity requirements for satellite communications.

From the point of view of the satellite provider, it is an uniform data stream, with specific restrictions but, nevertheless, a stream with a constant throughput requirement. The provider can plan and provision resources in advance for this service. On the other hand, emerging services using satellite communications present distinct challenges. Such applications, like satellite-based internet access or more recent interactive services, i.e., Video-on-Demand (VoD), have different traffic characteristics from traditional ones. Let us consider satellite television as an example. While the conventional architecture does not present major difficulties, recent applications, like the aforementioned VoD, have generally uncertain throughput requirements over time, so the satellite network must support these bursty unscheduled users.

The introduction of multi-beam satellites allows for a significant increase in throughput [RDG04] for the same RF power and bandwidth. However, multi-beam satellites with conventional payloads provide an uniform resource assignment per beam, i.e., the same time servicing each beam and throughput is assigned to service each beam. Thus, the optimal operating point of the systems lies in an uniform throughput demand across all beams. The latter, however, is not a fair assumption in real systems.

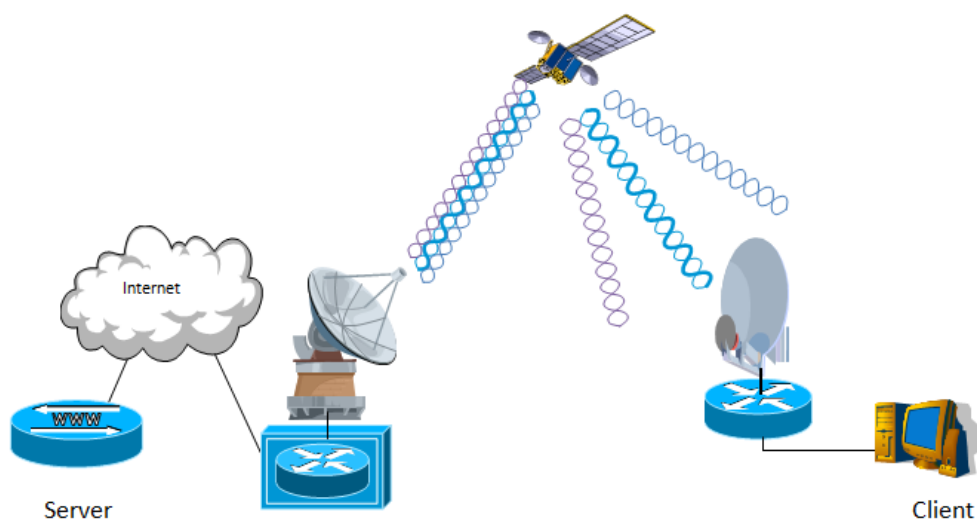


Figure 1.1: Typical satellite network topology for satellite-based internet access service. From left to right: ground station connected to the Internet, satellite acting as intermediary and end user station.

Furthermore, in multi-beam satellites a significant signal overlap between the different beams exists. A fraction of the terminals can, thus, simultaneously receive data from two or more beams, resulting in the existence of alternative paths to those terminals.

The main problem of current satellite networks is the lack of flexibility in the adaptation to changing traffic demands and to the inherent randomness of Internet traffic, resulting in the misuse of the satellite's resources. In view of this situation, we can formulate the problem at hand as the need for a more efficient use of the satellite's resources, with various objectives in sight, e.g., guaranteeing Quality-of-Service (QoS), taking advantage of alternative paths available and flexible coding techniques.

Flexible satellite payloads with adjustable power and bandwidth per beam or adjustable illumination time per beam (beam hopping) [ACG<sup>+</sup>10] allow the system to overcome the limitation of uniform resource allocation. It also addresses the issue of unequal traffic demands [ACG<sup>+</sup>10] across geographic regions. Other power allocation schemes [HSC<sup>+</sup>08] or beam-forming techniques [SB04] allow for multi-beam satellites to implement demand based capacity allocation between beams.

All these approaches focus on delivering flexible payloads for next generation satellites without addressing the inefficiencies of existing systems with conventional payloads. Besides, these resource allocation strategies i) have a allocation granularity that is coarse, and ii) are designed based on long-term statistics of the traffic and channels, making more refined and adaptive mechanisms for resource allocation to be instrumental to improving system performance for rapidly changing traffic loads.

Capital for performance improvement in front of rapidly changing traffic loads are the flexible coding techniques, such as network coding. In fact, with network coding, each route is not required

to send all data packets through two paths, as is usually the case for conventional soft-handover mechanisms, but rather a fraction of the original data only. Reduced unnecessary traffic results in less resources consumed.

### 1.2 Objectives

The main goal of this thesis is to increase serviceable demands by leveraging multiple available routes, resulting from beam overlap, in multi-beam satellite networks and flexible coding techniques.

To achieve this goal, we shall design and characterize load balancing algorithms that take advantage of the multiple available routes to each terminal, with the objective of increasing serviceable demands and reducing the delay that the data packets experience in the system. The alternative routes are currently untapped resources that offer an additional degree of flexibility in resource allocation. For this reason, we expect in our work to find improvements in the satellite's system throughput.

Flexible coding techniques, such as network coding, shall be used as they provide additional dimensions and advantages that may prove instrumental in the design of routing algorithms in multi-beam satellite systems. Likewise, it has been shown that network coding improves the communication's performance and reliability [FLBW06].

Altogether, we propose to design and implement the allocation algorithms and evaluate their performance. In order to perform the evaluation, a custom simulator is developed which allows to test the algorithms under different conditions, e.g., traffic patterns and intensity. Additionally, we analyse the relevance of coding techniques in this context, as for providing redundancy.

### 1.3 Structure

This report has two chapters more besides this introduction. Chapter 2 will present the state-of-the-art revision, that is, the research done in the area related to the thesis.

Chapter 3 presents the problems explored in this project. Firstly we detail the mathematical modelling of the routing problem. We then move to the setting in which the algorithms operate, introducing the relevant notions necessary to the working of the algorithm.

Chapter 4 discusses the algorithms developed to tackle the problems. In this chapter, we detail the design of the algorithms developed during this work.

Chapter 5 firstly gives an overview of the simulation model used throughout the project. The used metrics are described and then we assess the benefits brought by the algorithms by evaluating them in various situations, with the developed simulator.

Finally, the last chapter draws the major conclusions from this work and analyses the possible future work.

## Introduction

# Chapter 2

## State of the Art

### 2.1 Introduction

This chapter focuses on the bibliographic revision, discussing related works in order to present the existing state of the art of this field. Topics such as multi-beam satellites and network coding are addressed and discussed, and form the basis for the thesis work.

Multi-beam satellites introduce not only new opportunities but new challenges as well. Coupled with new allocation techniques, multi-beam satellites systems can overcome the limitations of uniform resource allocation. Still, these resource allocation strategies are based on long-term traffic patterns. We propose to address these inefficiencies, improving performance when the system is faced with rapidly changing traffic demands.

We present in this chapter relevant technologies that are already being used in the context of multi-beam satellites and others which we find provide additional benefits. We detail concepts such as multipath routing, which is associated with load balancing. Additionally, we discuss network coding techniques, which applied in this context constitute a novelty. Network coding plays an important role in the scope of multipath routing.

### 2.2 Multi-beam satellites

#### 2.2.1 Introduction

Most commercial satellites currently support multiple spot beams [HSC<sup>+</sup>08]. These satellites use multiple narrow spot beams in its coverage area as opposed to a broad beams. The satellite's coverage area is divided in cells, each served by an individual beam, as shown in Figure 2.1. The various regions and, therefore, the various beams have different demands, changing over time. This diversity is noticeable in Figure 2.2, which shows the assumed traffic distribution [ACG<sup>+</sup>10].

In order to guarantee flexibility in resource allocation, dynamic power allocation techniques to each beam have been used [ACG<sup>+</sup>10]. Power allocation schemes [HSC<sup>+</sup>08, CC05, DP11] or beam-forming techniques [SB04] allow for multi-beam satellites to implement demand based bandwidth allocation between beams. These will be discussed in more detail in section 2.2.2.

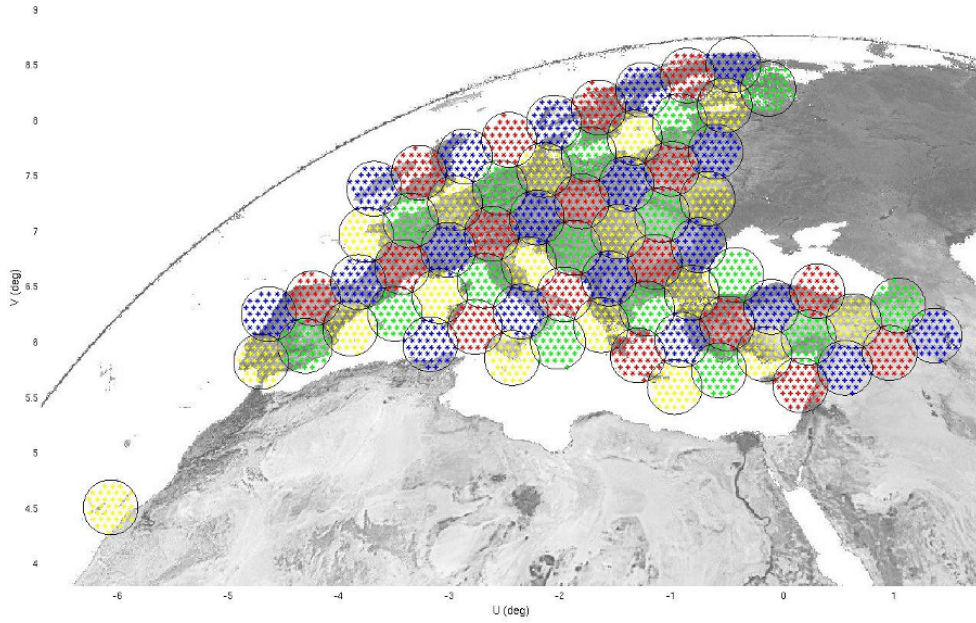


Figure 2.1: Coverage area and beam layout, displaying frequency reuse. Each cell represents the coverage area of a given beam; each colour a given frequency.

Furthermore, multi-beam satellite design is coupled with adaptive coding and modulation (ACM) allows for substantial capacity gains [RDG04]. This comprises the flexibility to adapt to changing channel and environmental conditions. It acts therefore as a means to maintain quality of service per delivered data packet. ACM systems improve the rate of transmission and bit error rates. ACM consists in the matching of the modulation, coding and other signal and protocol parameters to the conditions on the radio link. The radio link conditions may change due to interference from extraneous transmitters, for example. Adaptive modulation systems imply a channel state information at the transmitter. In time division duplex, this could be acquired by assuming the outbound and inbound channels are similar.

One measure of the condition of a radio link is the Signal to Interference plus Noise Ratio (SINR), which is defined as following:

$$\text{SINR} = \frac{P}{I + N}, \quad (2.1)$$

where  $P$  represents the received power,  $I$  the interference power and  $N$  the background noise. SINR is commonly used as a way to measure the quality of wireless channels. A wireless communication network needs to take environmental parameters into account to establish the quality of the link, SINR seeks to represent this aspect.

Associated to the SINR is the spectral efficiency, that is defined as the information rate that can be transmitted over a given bandwidth, in this case, in a satellite communication system. It is expressed as bit/s/Hz. For a given satellite channel, the bandwidth can be assumed to be constant

## State of the Art

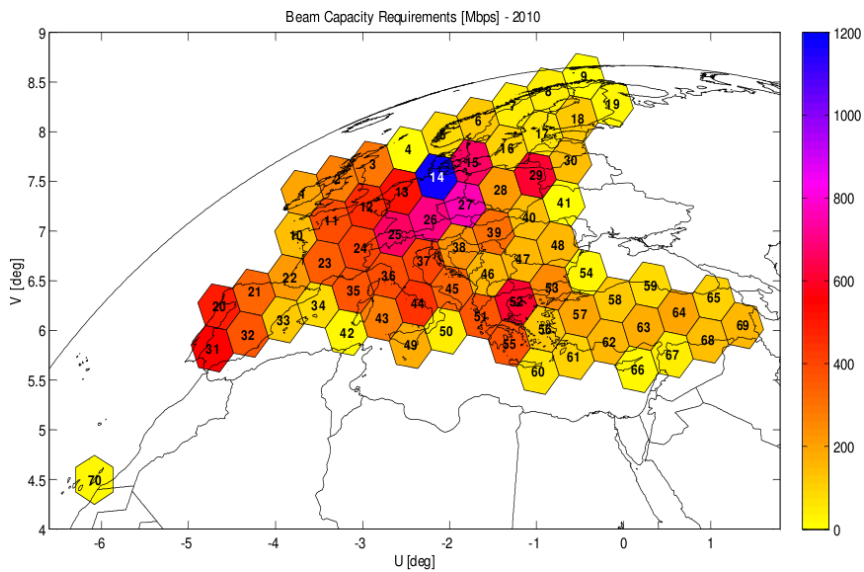


Figure 2.2: Assumed traffic distribution in 2010.

Source: [ACG<sup>+</sup>10]

so the spectral efficiency determines the channel's throughput.

### 2.2.2 Power allocation schemes and beam allocation techniques

A satellite's throughput is related to the power, by increasing the power the capacity is enhanced. However, the total power limit of the satellite restricts its total capacity. Increasing power leads to a better signal, improving the radio link and thus increased capacity. The requests generated by the terminals within a given beam constitute the aggregated traffic demand of that beam [HSC<sup>+</sup>08]. The aggregated demand is the factor the resource allocation algorithm takes into account.

Several current satellite systems have the same bandwidth and power allocation for all the beams. The fixed beam power allocation translates in an eased resource allocation problem and in a simplified receiver station implementation [RDG04].

In order to achieve capacity re-allocation, various methods have been devised such as the use of flexible TWTs and multipoint amplifiers that provide variable power per beam, and analogue or digital processors to assign bandwidth per beam on demand. Along with beam-forming technique, which is a signal processing technique for directional signal transmission. User-specific beams can be formed by coherently controlling the base station antennas. Additionally, individual beams can have their SINR controlled by adaptively controlling the beams, when channel information is available. These techniques allow for multi-beam satellites to implement demand-based bandwidth allocation between beams [SB04].

An alternative method with its own set of advantages is beam hopping. Beam hopping is a technique in which only a subset of the satellite's beams are illuminated at a given time. This

allows for flexible management of resources, as more capacity can be allocated to beams according to traffic demands, over the satellite's lifetime [ACG<sup>+</sup>10].

When using the beam hopping technique, every beam is not illuminated simultaneously, but only a subset of beams is illuminated at a given time. This is effectively the timesharing of active downlink beams [HSC<sup>+</sup>08]. Hence, a time and spatial transmission plan, with a certain "window" length, is defined. Selected beams have access to the spectrum during its time slot.

Naturally, beam hopping offers additional flexibility as opposed to fixed power allocation. Nevertheless, illumination schedules are based on long-term traffic statistics for the various covered regions. Hence, beam hopping does not offer the flexibility to adapt to rapidly changing demand or sudden spikes in traffic.

### 2.2.3 Challenges

All these approaches focus on delivering flexible payloads for next generation satellites without addressing the inefficiencies of existing systems with conventional payloads. Furthermore, they are typically designed for long term demands on the system without considering the short term load imbalance created by traffic dynamics.

Ground stations on the border of a cell may be able to receive transmissions from beams assigned to neighbouring cells. This should enable the satellite to transmit data to a given station through more than one beam, resorting to multipath routing. This field has not been particularly explored and, yet, it would be a further instrument in increasing the satellite's serviceability.

In a multi-beam satellite, no single entity controls all the beams. Instead, a set of Gateways to Users provide the service. The gateways are linked to the downlink beams in a multi-star configuration in which a Gateway accesses a fixed set of beams [ACG<sup>+</sup>10]. As coordination among gateways is minor or nonexistent, a terminal that is covered by beams from two different gateways is limited to only one of them and does not have alternative paths. This is a hurdle whose impact may be overcome by the overall benefits of multipath routing.

As stated previously, the several studies mentioned do not address the inefficiencies of actual systems with conventional payloads. We propose to explore the potential benefits of the multiple available paths to send packets to a given user, allied with a dynamic allocation of resources.

## 2.3 Multipath routing

Multipath routing is a routing technique that resorts to multiple alternative paths in a network. This routing allows for improved security, reacting to failures, and load balancing [HR08]. Multipath routing allows the system to adapt to application performance requirements, using paths with different characteristics to cater for different needs. For example, the system can provide a low-delay or high-throughput path depending on the user's needs. An improvement in reliability is achieved due to the potential to use alternative paths in case of failure.



All in all, multipath routing protocols seek to take advantage of the path diversity in a network. They do in fact, with the appropriate characteristics, enable load balancing across several paths, bringing network robustness and reliability [MPC08].

In [CRS99], the authors analyse the performance of multipath routing algorithms that reserve resources along the paths that are considered for routing. They compare the performance of multipath algorithms with single-path algorithms, in terms of throughput and the delay in connection establishment. They show that multipath routing presents benefits for bursty applications.

Several works have been undertaken related to multipath routing in wireless networks ([ZL09, RGKR08, HZL<sup>+</sup>08, ZL08]). However, multipath routing in the context of satellite communication has not been widely explored.

In [RGKR08], the authors identify the major factors a system needs to take into account in order to use the resources of a wireless mesh network efficiently. Firstly, it must account for the existence of multiple paths. The broadcast nature of the channel, including the existence of multiple transmission powers and rates, must be observed as they affect the probability of correct packet reception. Lastly, the system must take into account the competition among multiple data flows. The authors highlight the importance of optimizing all the above parameters simultaneously, to take into account the broadcast nature of wireless systems and address fairness issues, to achieve optimal performance.

In [SBJ<sup>+</sup>11], the authors present techniques for achieving high throughput in multihop wireless networks. They analyzed, specifically, transmissions via satellite to a mobile terrestrial user, exploring strategies to make efficient use of bandwidth in this conditions, i.e., a network where instantaneous capacity can vary by orders of magnitude in an unpredictable manner. They demonstrate that network coding can provide high throughput, delay-constrained communication while utilizing bandwidth as efficiently as single-path routing. Network coding, and the benefits it brings in the context of multipath routing, will be discussed in detail in the next section.

## 2.4 Network Coding

### 2.4.1 Definition

Network Coding is a recent field in information theory that generalizes the means of conveying information in communication networks, breaking with classical mechanisms that have ruled operation of networks. Instead of simply forwarding data, nodes in the network may recombine several input packets into one or more output packets [FLBW06]. This requires enhanced computational capabilities from the network's nodes but that is not a problem as processing is becoming less expensive. The problem is rather the network bandwidth. Network coding increases network efficacy by means of cheap computational power.

Conventional wisdom states that for multicasting, there must be a logical connection among all the parties involved. This translates into the replication of bits of the message at the intermediary node, so that each sink node finally receives a copy of the message. It has been shown that the

traditional technique for multicasting in a computer network is generally not optimal [ACLY00]. An example of multicasting is shown in Figure 2.3.

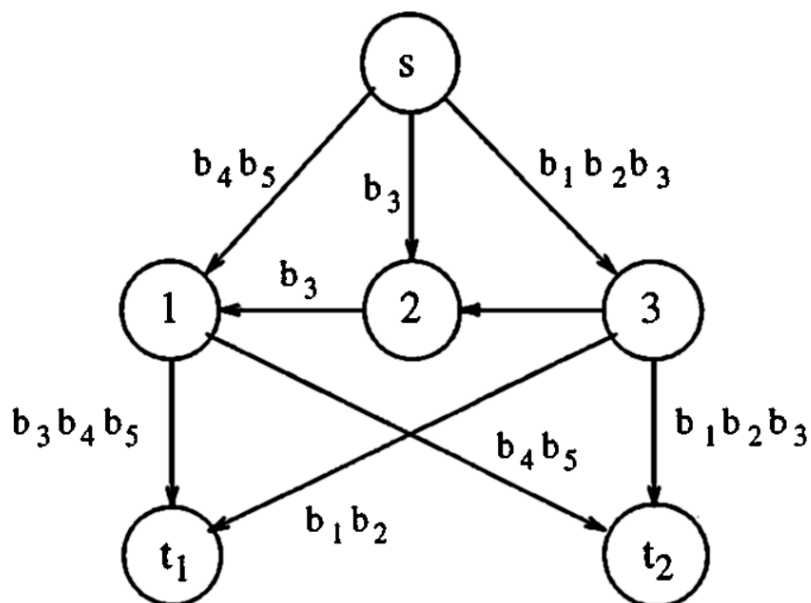


Figure 2.3: Example of traditional multicast: the central nodes receive different packets from the source node. Each sink node receives the packets through different paths.

Source: [ACLY00]

Rather than simply forwarding incoming packets, nodes can recombine several input packets into one or more output packets. This is done in practice by a linear combination of data: network coding protocols interpret data packets as numbers over some finite field [FLBW06]. Nodes in the network forward linear combinations of these symbols across the network, thus leveraging basic features of linear codes such as erasure correction capability and well understood encoding (linear combination) and decoding algorithms (e.g., Gaussian elimination). An example of network coding is shown in Figure 2.4. The figure depicts the successive transmissions, from top to bottom: left and right nodes transmit packets A and B, respectively. After receiving the two packets, the middle node broadcasts a single packet, combination of the two it received. The nodes are able to decode this coded packet using, each, the packet they sent before. In the end every node has received the packets A and B.

Previous studies have shown that linear coding is sufficient to achieve the multicast capacity ([KMM03, LYC03]). [HMK<sup>+</sup>06] demonstrated that Random Linear Network Coding (RLNC), that is, the use of randomly generated linear codes in a distributed manner also achieves capacity with high probability.

The procedure of generating a linear coded packet is shown in Figure 2.5. We are assuming two original packets of the same length:  $n$  bits. Considering operations over symbols of  $g$  bits, so, operating on a Galois Field  $\text{GF}(2^g)$ , two symbols ( $C_1, C_2$ ) are chosen to represent the coefficients of the linear combination. Each packet is split into symbols of  $g$  bits and each symbol of packet  $P_i$

## State of the Art

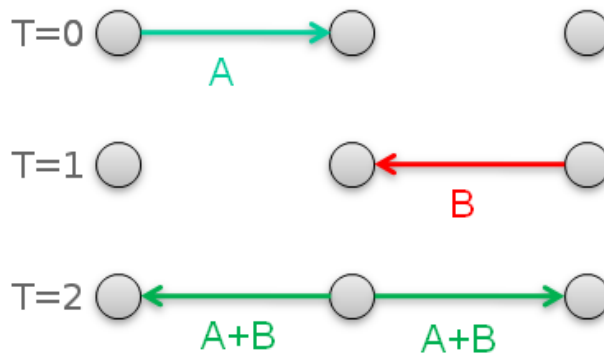


Figure 2.4: Example of network coding: the middle node receives packets A and B. It then broadcasts a packet, result of the combination of the incoming packets. The left and right nodes can recover the missing packet using the one they sent and the coded packet they receive.

is multiplied by the coefficient  $C_i$ . These products are added for the same symbol position of each packet. These symbols constitute the data content of the coded packet, which is concatenated with a header and the coefficients  $C_i$  used in the generation of the coded data. These latter are needed for the recovery of the information at the receiver end.

### 2.4.2 Benefits

Network coding has been demonstrated to deliver gains in many scenarios, such as improving reliability in lossy wireless links [FLBW06]. It has been shown that network coding techniques can provide robust multicast over lossy satellite channels and that it can be implemented at encapsulation level (GSE) [VB09], thereby making it transparent for IP packets and to the network outside the satellite segment.

This allows for improved efficiency in comparison to Forward Error Correction (FEC) techniques [VCL08], that are used for controlling errors in data transmission over unreliable or noisy communication channels. They are typically used as complement to Fade Mitigation Techniques (FMTs) [LBAMG07] for mobile satellite terminals. However, network coding can also be used to deliver coded packets over multiple routes, especially over lossy wireless links [ZL09], [ZL08], without investing additional resources in repeating the same packets through each route for enhanced reliability. Our work focuses on algorithms that specify the transmission rates for each route as a consequence of providing load balancing across beams.

As mentioned previously, network coding improves the reliability of a lossy channel. Traditionally, dealing with packet losses involves the use of erasure codes, that consist in transmitting redundant data. The purpose being that a portion of the packets can be lost and the receiver still be able to reconstruct the original data. Without network coding, the redundant packets are simply copies of particular packets. So to recover a lost packet, a copy of that specific packet has to happen to have been transmitted and, naturally, received successfully.

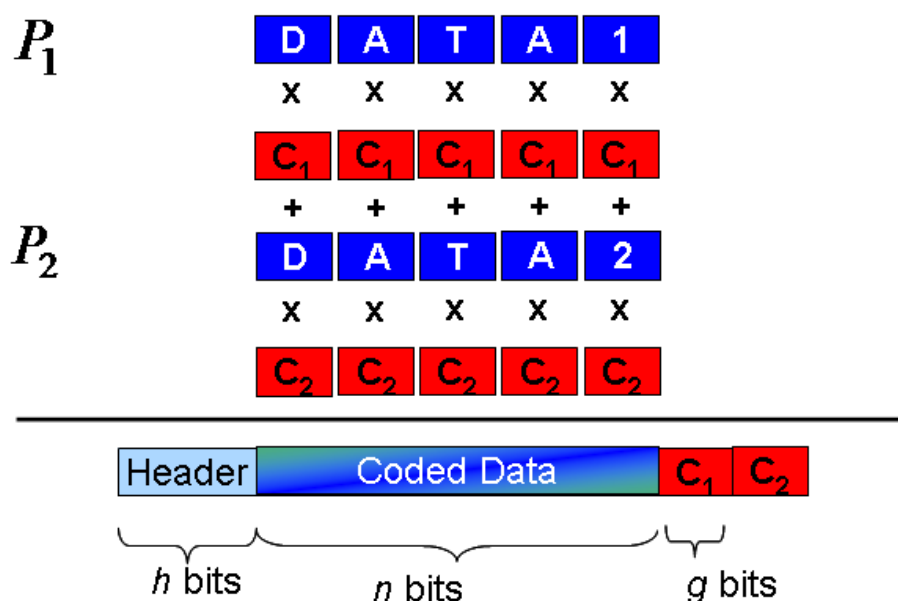


Figure 2.5: Example of linear combination of two data packets using network coding. We use symbols in  $GF(2^8)$ , data packets are of  $n$  bits,  $C_i$  represents the coefficients of the linear combination. The process ends in a coded packet, composed by coded data, a header and the encoding coefficients.

With network coding, a set of coded packets are sent, considering that for  $N$  original packets,  $N + M$  coded packets are sent. All of the original packets can be decoded if no more than  $M$  packets are lost. However, if not, no packets can be decoded. As reported in previous studies [PR11], this strategy, called Dense Network Coding (DNC), is optimal in a lossless channel but not recommended in a lossy channel. In a channel with low loss probability, this strategy can be even worse than no redundancy at all.

Several studies have been published suggesting an alternative strategy to avoid the DNC problems — Systematic Network Coding (SNC) [LMaS10, PR11]. For each batch of  $N$  packets,  $N + M$  packets are sent, but the first  $N$  packets are the original ones. The reception of each these packets is independent of the reception of other packets, since no additional packet is necessary to decode it. The remaining  $M$  packets are the redundant ones, being randomly generated combinations of the original packets. This way, a packet is successfully delivered if (i) the uncoded packet is received or (ii) otherwise, but at least  $N$  of the remaining  $N + M - 1$  packets are received.

### 2.4.3 Applications

As mentioned above, network coding has the potential to provide for reliable multicast over satellite networks by offering a native and transparent solution. Several studies investigating the benefits and applications of network coding in multicast networks have been published ([ZL08, HMK<sup>+</sup>06, VB09]).

## State of the Art

As investigated in [VB09], the developed framework has benefits in efficiency improvements in reliable multicast from the lack of FEC overhead, eliminating redundant data and very low decoding delay as the channel diversity minimizes redundancy in the recovery data. It allows for transparent multicast based on network coding approaches. It is pertinent that this framework is applicable to both unidirectional and bidirectional satellite systems and to fixed and mobile terminals. The work in [VB09] demonstrated performance improvements between traditional and network coding approaches, pushing performance close to the theoretical bounds.

In [ZL08], the authors address the problem of resolving conflicts of interest among multiple competing flows with wireless multipath network coding. They model the problem as a network game, in which the participating players share the bandwidth resource through negotiation or competition. Their work consists in an attempt to apply game theory to multiple competing flows in the context of wireless network coding. From the modelling of the problem, a localized optimization of two sub-problems is performed: multipath opportunistic routing and coding rate allocation.

In [YY08], the authors' routing protocol, that relies on network coding, was found to enhance the reliability of Wireless Sensor Networks (WSN). A WSN consists of spatially distributed autonomous sensor that cooperatively transmit their data through the network to a central location. It is shown that the original packets can be recovered from packets with chaotic order or partial information loss.

Other works have investigated the benefits and trade-offs of applying network coding to multicast ([HZL<sup>+</sup>08, SBJ<sup>+</sup>11, PMaSO10]). These studies report that network coding can provide high throughput and delay-constrained communication. Specifically regarding multicast, their results show that network coding is nearly as effective as flooding the network, but still utilizing bandwidth nearly as efficiently as with single-path routing.

## State of the Art

## Chapter 3

# Problem Description

### 3.1 Introduction

The satellite system is charged with transmitting data packets from and back to earth stations. It is subjected to pressing restrictions, in terms of available power and temporal tolerances regarding packet transmission delays. One noted drawback of current systems is the relatively poor performance of the system when faced with uneven and unpredicted traffic patterns.

Currently deployed technology does not offer enough flexibility to accounting for rapidly varying demands. We propose to tackle this problem by exploring unused paths provided by neighbour beams. In conventional systems, each terminal is assigned to a particular beam, through which its whole traffic is routed. By moving a portion of a session's packets to a neighbour beam we expect to use otherwise unused resources and reduce transmission delay.

### 3.2 Methodology

This thesis starts out by laying out the mathematical formulation of the problem, that is, the mathematical model of multi-beam load balancing. From there we develop algorithms for coding-aware multipath routing, taking advantage of the multiple paths available. Another matter we worked on is the development of algorithms for the allocation of resources. Furthermore, we also develop a simulator for the characterization of the algorithms.

#### 3.2.1 Mathematical formulation of multi-beam load balancing

Let  $B$  be the set of beams  $B$  and  $K$  a set of sessions. Each beam  $i \in B$  supports a set  $K_i \subset K$  of unicast sessions. We assume a session is assigned to a beam because that beam provides the highest SINR. We define  $J_{ij}$  as the set of sessions assigned to beam  $i$  but that may be served with beam  $j$ . Each beam  $i$  has a set of  $N(i)$  neighboring, i.e., overlapping, beams.

Define  $R_{s_l}^k$  as the transmission rate of session  $s_l$  when served by beam  $k$ , which depends in part on the achievable MODCOD rate. Let  $M_{s_l}$  and  $N_{s_l} \geq M_{s_l}$  represent the number of original and coded packets of session  $s_l$ , respectively, where the latter constitutes the number to be sent

## Problem Description

by the system.  $T^i$  represents the transmission time of beam  $i$  when operating independently from overlapping beams, i.e., servicing all sessions assigned to it. Thus,

$$T^i = \sum_{s_l \in K_i} N_{s_l} / R_{s_l}^i, \quad (3.1)$$

if network coding is used. In a scenario with no coding,  $N_{s_l}$  should be substituted by  $M_{s_l}$ . Finally,  $T_i$  is defined as the transmission time for beam  $i$  when cooperation, i.e., resource allocation, is allowed amongst neighbouring beams, where  $\alpha_{s_l}^i$  constitutes the fraction of the packets from session  $s_l$  delivered by beam  $i$ . Naturally,  $\alpha_{s_l}^i + \sum_{k \in N(i)} \alpha_{s_l}^k = 1$  for every session  $s_l$  assigned to each beam  $i$ .  $T_i$  is determined by the  $i$ th beam's contribution to three types of sessions:

- i. assigned sessions that cannot be covered by any other beam,
- ii. assigned sessions partially serviced by other beams, and
- iii. sessions assigned to other beams that the  $i$ th beam partially covers.

These effects correspond to the three terms in the following equation:

$$T_i = \sum_{s_l \in K_i, s_l \notin J_{ij}, \forall j \in N(i)} \frac{N_{s_l}}{R_{s_l}^i} + \sum_{j \in N(i)} \sum_{s_l \in J_{ij}} \alpha_{s_l}^i \frac{N_{s_l}}{R_{s_l}^i} + \sum_{j | i \in N(j)} \sum_{s_l \in J_{ji}} \alpha_{s_l}^i \frac{N_{s_l}}{R_{s_l}^i}. \quad (3.2)$$

where  $N_{s_l}$  is the number of packets from a given session to be transmitted,  $R_{s_l}^i$  is the transmission rate for that session in beam  $i$ . This equation determines the beam's transmission time based on the routes selected for the sessions. The transmission time is key in assessing whether a specific traffic allocation is valid.

The equations presented in this section are used in a linear programming problem, whose optimization and solutions found contribute to benchmarking our algorithms. The variables present in the model are  $\alpha_{s_l}^i$  — the fraction of packets from session  $s_l$  delivered by beam  $i$ . The model has three types of restrictions: i) the different paths available in each beam, ii) the sum of the fractions of a session's packets must be 1, and iii) a beam's transmission time cannot exceed the maximum allowed. Finally, the optimization seeks to minimize the global transmission time.

### 3.2.2 Algorithms for multipath allocation

#### 3.2.2.1 Network coding applied to multipath routing

The use of multiple beams to communicate with a ground station implies multipath routing. In the case of multi-beam satellites, this routing is not so complex as there are no intermediary nodes present: just the source and destination nodes with multiple paths linking them. Taking a terminal that can listen to two beams from a multi-beam satellite, the satellite can send data through these two different beams and, therefore, two different routes.



## Problem Description

A traditional system would perform a soft-handover, that is, would assign resources in both beams and send the same packets through both of them. This decision ignores the beams' load from other terminals and is meant to provide additional reliability.

We can oppose network coding in multiple route scenario to a more traditional soft-handover technique. In a network coded system, the system has the flexibility to send distinct fractions through the different beams, provided that enough coded packets are sent. It has also the ability to decide how much redundancy to the transmission should include. Thus, a system might send more packets through a given beam because of channel constraints or capacity saturation on another beam. By sending different linear combinations through each route, a coded system can provide additional guarantees of recovering data packets, as opposed to a traditional system. Additionally, by sending distinct fractions of the data batch through the different paths, the system is able to save resources.

In RLNC, with a large enough field size, coding across  $M$  packets requires  $M$  coded packets to be received at the end terminal in order to recover all original packets. This is a key challenge to analyse as part of this thesis. There must be found a trade-off between partial recovery of data by using a sparser code, as RLNC is a dense code, and the inherent loss in performance due to the sparser nature of the code — RLNC is delay-optimal for a large enough field. Systematic network coding provides no degradation in performance while ensuring: i) partial recovery of the packets, and ii) a reduction in decoding complexity, as shown in [LMaS10].

Previous work on network coding for lossy satellite channels focused on techniques that provide robust multicast [VB09]. Network coding can be implemented at the encapsulation level, thus making it transparent for IP packets and to the network outside the satellite segment. This allows for improved efficiency in comparison with FEC techniques [VCL08] that are typically used as a complement to fade mitigation techniques for mobile satellite terminals [LBAMG07].

### 3.2.2.2 Resource allocation for reliable provision for a single user

Let  $M$  be a set of packets  $m_1, \dots, m_N$  to be delivered from source node  $S$  to destination node  $D$ . There are  $K$  paths available, having each path an associated error probability, denoted by  $e_i$ , in which case the packet is not delivered successfully. We want to perform the allocation of packets among the available paths, this allocation being denoted by  $m_1, \dots, m_i, \dots, m_N$ . This allocation must ensure a certain probability of successful recovery of the packets, denoted by  $Pr$ :

$$Pr = \sum \binom{m_i}{k_i} e_i (1 - e_i)^{K_i} \quad (3.3)$$

where:

$$\sum K_i \geq M \quad (3.4)$$

The aim is to have  $Pr \geq 1 - \varepsilon$  where  $\varepsilon > 0$ ; being  $\varepsilon$  the upper bound for the probability of not delivering enough packets.

## Problem Description

In order to meet this requirement, it has been shown that the number of additional packets can be obtained through equation 3.5 [ML12]:

$$N_2 = \frac{1}{p_2} \left( \beta \left( e_2 + \frac{1}{3} \right) + K' - N_1 p_1 \right) + \frac{1}{p_2} \sqrt{2\beta \left( N_1 p_1 (e_1 - e_2) + K' e_2 + \frac{\beta}{2} \left( e_2 + \frac{1}{3} \right)^2 \right)} \quad (3.5)$$

Where  $K$  is the batch size, that is, the session's number of packets to send;  $N_1$  and  $N_2$  the number of packets to be send through each path,  $e_1$  and  $e_2$  the channel's erasure probabilities and  $p_1$  and  $p_2$  the channels' success probabilities, being  $p_n = 1 - e_n$ ; where  $\beta = \log(1/\epsilon)$  and  $K' = K - 1$ .

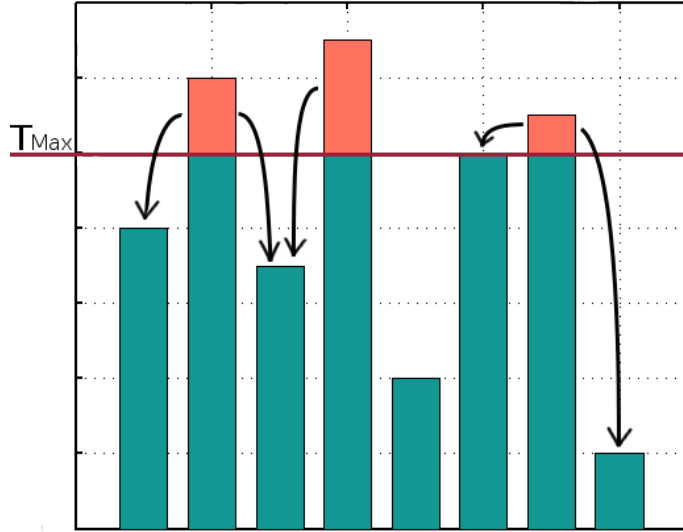


Figure 3.1: Example of load balancing:  $T$  is the time available during a frame and the bars represent the amount of time it would take to each beam to deliver its data. We can exploit other beams' unused resources.

From Equation 3.2 and Equation 3.5, packets are sent through alternative paths according to what each beam's load permits. An example is shown in Figure 3.1, in which three beams are saturated, that is, the time it would take them to deliver the data to their assigned users exceeds the time  $T_{Max}$  available during a frame. The algorithm can route some of the packets through neighbour beams that do not have so much traffic, effectively performing load balancing. The goal being to exploit the resources that have not been used by other beams. Likewise, the goal — and consequence — of load balancing is to increase the system's stable throughput region.

### 3.2.3 Algorithms for allocation of resources

It has been demonstrated that, in order to increase throughput, multi-beam satellites should allocate power dynamically as on-board resources are scarce and expensive [HSC<sup>+</sup>08]. The satellite has a

## Problem Description

maximum output power, denoted by  $P_T$ , being the output power given by:

$$P_T = \sum P_i \quad (3.6)$$

Where the power for  $i^{\text{th}}$  beam  $P_i$  is given by  $P_i = \gamma T_i$ , being  $\gamma$  a multiplier derived from the channel and environmental conditions.

The allocation of power, that is, the allocation of the illuminated beams must follow physical restrictions and the capacity demands for each beam. The specifics of power allocation and its algorithms are beyond the scope of this work. Herein, it will be assumed that these tasks are already covered. Thus, the developed algorithms will handle the allocation problem at a higher level.

One plain objective for the algorithm is that of increasing the system's overall capacity. In this way, every session is treated rather equally save for a possible advantage for the sessions with best transmission rates. Beyond this objective, other goals exist depending on the satellite operator's needs.

Alternative goals derive from eventual contracts or agreements, such as a determined Fair Access Policy (FAP). Such a policy commonly defines the connection's guaranteed bandwidth and may impose a bandwidth cap, limiting transmissions after a certain threshold. A policy might also set different traffic priorities. In this situation and with great load, some sessions may be more heavily delayed in order to allow for priority traffic to be transmitted without delay.

The resource allocation algorithm takes therefore the policy into account. It should establish the value of importance of each session. When faced with the need to delay traffic, it can then start off by delaying the least important sessions.

### 3.3 Conclusions

In this thesis, we address the current limitations, notably the lack of flexibility the system has to accommodate different traffic patterns. We approach the unused resources by implementing resource allocation and coding-aware multipath algorithms, that improve packet broadcast by specifying transmission rates and achieve load balancing. The work will be carried out attending previous research done in this field and taking advantage, particularly, of network coding techniques that have not been widely explored in the context of multi-beam load balancing.

The algorithms designed following the mathematical formulation of the problem are the focus of this thesis. A simulator shall be developed to be able to, then, characterize the algorithms, in a controlled environment. It is our belief that this will constitute a positive contribution to this ever evolving area. Its applicability in the real-world commercial satellites makes it the more useful.

## Problem Description

# Chapter 4

## Algorithms

### 4.1 Introduction

In this chapter, we firstly describe the algorithms for the allocation of traffic. The algorithm performs a static allocation, that is used to verify whether additional steps are needed. If they are indeed necessary, it also acts as a preparation for them.

The following section presents the load balancing algorithm. The algorithm's steps are detailed and additional matters are approached. We analyse two different algorithms: the static allocation and the load balancing algorithm. The latter is central in this thesis.

Finally, the last section discusses the packet discard algorithm, which works in conjunction with the previous algorithms. In this work, the additional business dealing with traffic policies are left to the discard algorithm. The last section weights on those matters.

The system is composed by a number of user sessions and a fixed number of beams. Each session is assigned to a given beam but may be partially served by other beams. Thus, each pair session – beam constitutes a path with an associated transmission rate and an erasure probability, that is, the probability that a packet is not transmitted successfully.

One important metric is the beams' transmission time. The transmission time for each beam is based on the number of packets transmitted and the transmission rate. A specific allocation is valid if, and only if every beam's transmission time is not over its defined limit.

### 4.2 Static allocation

The process starts off with new incoming packets for each session, at the beginning of each transmission window. Before proceeding to the main steps, an initial configuration must be generated. The first step consists in allocating, for each session, the whole traffic to the best path. The best path is usually the one through the assigned beam for that session as that should be the one with the highest SINR and, thereby, with the best transmission rate.

The initial configuration is obtained in the following way, for each session:

## Algorithms

1. find the best path – with the best transmission rate – and allocate the entirety of the packets to that path.
2. Alternative paths carry no traffic.

This allocation ensures that the lowest global transmission time is obtained, as the best paths — with the best data rates — are used. This configuration consists, in fact, in the expected allocation if there were no alternative paths available, as one path alone is used per session.

This configuration is optimal in the case of low enough demand which we assume to lead to a valid configuration where every beam's transmission time is under its limit. The configuration is optimal due to the use of the best paths, which ensures the lowest transmission time and thus, the least energy consumption. In the instance of low demand, it is, thus, straightforward to obtain a valid and optimal configuration.

If the configuration is not valid, additional steps are needed. Without load balancing, there is only one option: the system must discard packets from the problematic beams' sessions, the beams whose transmission time is over the limit. As will be described in the following section, performing load balancing may reduce the number of discarded packets or drop the need to discard all together.

### 4.3 Load balancing

As discussed previously in this document, the key factor in making the most of satellite resources lies in the exploitation of currently unused but existing alternative paths.

As stated before, the static allocation performed grants the optimal allocation if the configuration is valid. Optimal allocation should be read as the best global transmission time. Thus, the load balancing algorithm starts from the configuration left by the static allocation. Performing load balancing reduces the number of discarded packets but every step increases the global transmission time. Therefore, the algorithm is designed to finish as soon as it reaches a valid configuration.

Prolonging the algorithm's run beyond the strictly necessary to reach a valid configuration can be misleading. On the one hand, the different beams' loads have the tendency to even out, and every beam to have a similar margin for future additional traffic. On the other hand, traffic is dispersed further, a fact that can deceive resource allocation algorithms into guessing a particular beam is less loaded. Thus, the algorithm effectiveness may be hindered.

#### Outline

The load balancing algorithm is divided into two very suchlike phases. The first phase is outlined by Algorithm 1, and can be summarized by the following:

1. Find a beam with too much traffic.

## Algorithms

2. For each session, move the maximum possible number of packets to another path, so that either (i) the first beam becomes saturated, (ii) the second becomes saturated, or (iii) no more packets can be moved in this session.

---

**Algorithm 1** Load Balancing pseudo-code.

---

```

for all Beams  $b1, b2$  do
  if  $\text{transmissionTime}(b1) > \text{maxT}(b1) \wedge \text{transmissionTime}(b2) < \text{maxT}(b2)$  then
    for Swap  $swap, swap.b1 = b1 \wedge swap.b2 = b2$  do      ▷ Swap between the two beams.
       $s1 \leftarrow swap.session$ 
       $trans \leftarrow \min(\text{xsl}(b1, s1), \text{xsl}(b2, s1))$ 
      if  $trans == 0$  then continue
      else
         $\text{xsl}(b1, s1) \leftarrow \text{xsl}(b1, s1) - trans$ 
         $\text{xsl}(b2, s1) \leftarrow \text{xsl}(b2, s1) + trans$       ▷ Move  $trans$  packets to beam  $b2$ .
        break
      end if
    end for
  end if
  if validConf() then
    break
  end if
end for

```

---

The second phase is outlined by Algorithm 2, and can be summarized by the following:

1. Find a beam with too much traffic.
2. Search a beam that has a session in common to use as an intermediary beam, and a third beam which, in turn, has a session in common with the intermediary beam.
3. While keeping the intermediary beam saturated, move the maximum number of packets from the saturated beam to the intermediary and from the intermediary to the third one, so that either (i) the first beam becomes saturated, (ii) the third beam becomes saturated, (iii) no more packets can be moved in the first session, or (iv) no more packets can be moved in the second session; whichever comes first.

The processes occurring in these two phases are illustrated in the following paragraphs. A concrete case is used as an example of a possible situation the algorithm has to face. Firstly, let us consider the situation presented in Figure 4.1, where we have 3 beams. A set of sessions exist for each beam, we highlight the purple and yellow sessions, being the other sessions treated aggregated as the blue sessions. Beam 1 is over the maximum transmission time and has a session in common with the second beam.

The algorithm identifies the possible swap, the purple session from beam 1 to the second one. After performing the swap, as presented in Figure 4.2, beam 1 is still over the maximum. The

---

**Algorithm 2** Load Balancing pseudo-code, phase two.

---

```

for all Beams  $b1, bInter, b3$  do
  if transmissionTime( $b1$ ) > maxT( $b1$ )  $\wedge$  transmissionTime( $b3$ ) < maxT( $b3$ ) then
    for Swap  $swap, swap.b1 = b1 \wedge swap.b2 = bInter$  do
      for Swap  $swap2, swap2.b1 = bInter \wedge swap2.b2 = b3$  do  $\triangleright$  Swap between the two
      pairs of beams.
         $s1 \leftarrow swap.session$ 
         $s2 \leftarrow swap2.session$ 
         $trans \leftarrow \min(xsl(b1, s1), xsl(bInter, s1))$ 
         $trans2 \leftarrow \min(xsl(bInter, s2), xsl(b3, s2))$ 
         $trans, trans2 \leftarrow \text{calculate}(trans, trans2)$ 
        if  $trans == 0$  then continue
        else
           $xsl(b1, s1) \leftarrow xsl(b1, s1) - trans$ 
           $xsl(b2, s1) \leftarrow xsl(b2, s1) + trans$   $\triangleright$  Move  $trans$  packets to beam  $bInter$ .
           $xsl(bInter, s2) \leftarrow xsl(bInter, s2) - trans2$ 
           $xsl(b3, s2) \leftarrow xsl(b3, s2) + trans2$   $\triangleright$  Move  $trans2$  packets to beam  $b3$ .
          break
        end if
      end for
    end for
  end if
end for
if validConf() then
  break
end if
end for

```

---

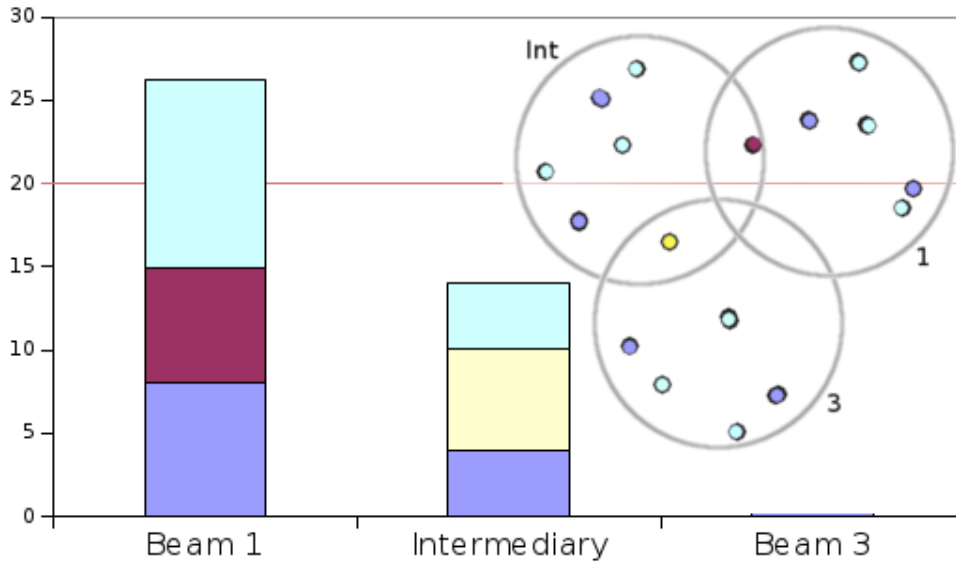


Figure 4.1: Diagram depicting the packet swaps performed by the algorithm in the second phase.



## Algorithms

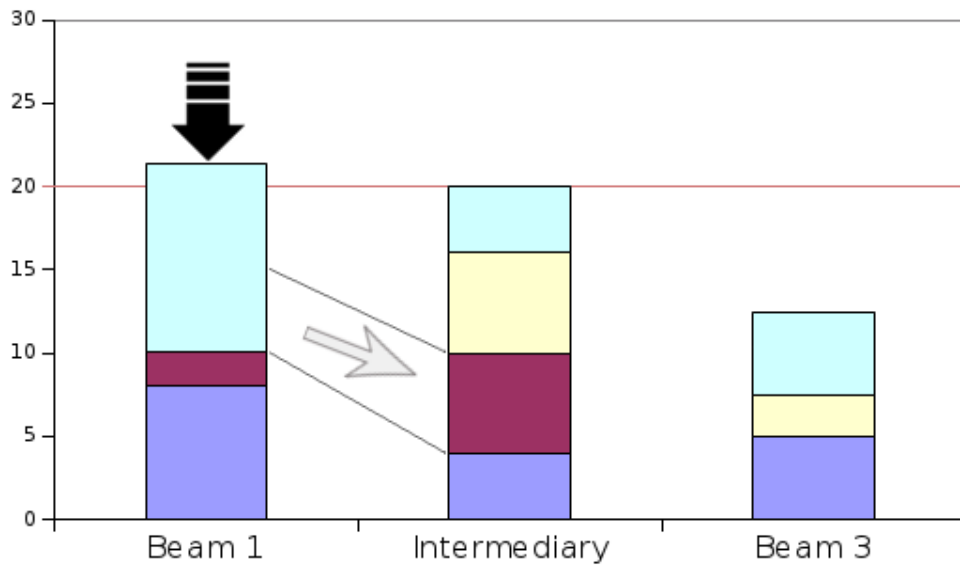


Figure 4.2: Diagram depicting the packet swaps performed by the algorithm in the second phase.

algorithm transfers the biggest amount of packets possible, in this case, until the second beam becomes saturated. At this point, no more swaps are possible so the algorithm proceeds to the second phase.

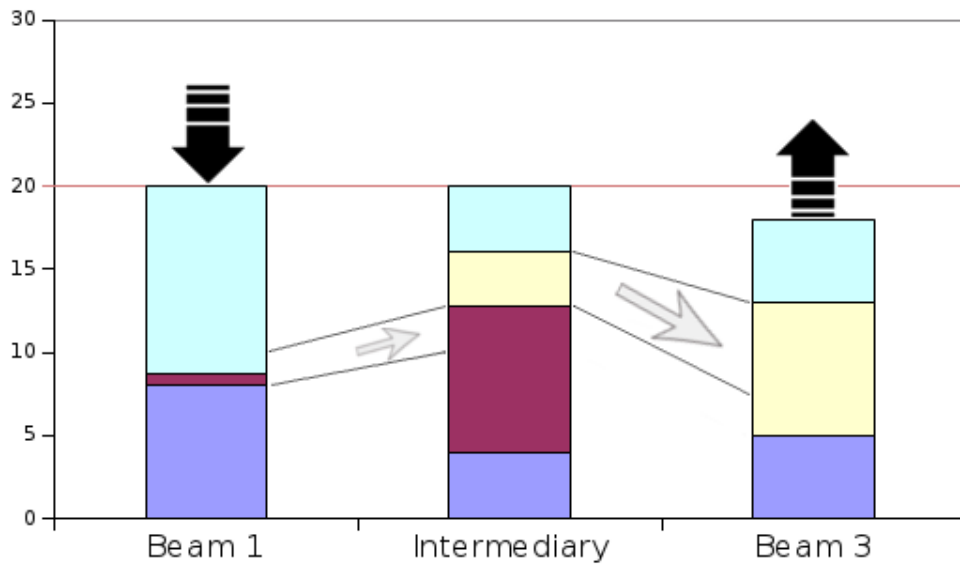


Figure 4.3: Diagram depicting the packet swaps performed by the algorithm in the second phase.

To reach the objective of offloading enough packets from beam 1, the algorithm tries to move packets from a neighbour beam to a third one. In this example, the algorithm identifies the yellow session. As shown in Figure 4.3, a fraction of the yellow session packets is moved to beam 3, as to allow further transfer from beam 1. A small enough number of packets is moved in the two swaps,

as to allow beam 1 to reach the maximum transfer time.

### Description

Should the configuration be invalid, the algorithm enters the first phase, moving some packets to neighbour beams. The neighbour beams provide paths with equal or worse quality, so the global transmission time increases inevitably. The ratio between the transmission rates in the original and neighbour beams determines by which factor the transmission time will increase after a load offset is performed.

The algorithm continues to the next phase if no more swaps can be found and the configuration is still invalid. The second phase is similar to the first but the algorithm cannot carry out direct swaps as the problematic beams does not have any non-saturated neighbour with a session in common. Instead of swapping packets directly from one beam to another, an intermediary is used to facilitate the process. The intermediary beam is a neighbour of the problematic and the final beams. As the intermediary beam may be saturated, in order to move packets there, it is important to also move from the intermediary to the third beam.

In this phase, the problematic beams have no neighbours available to offset their load. The algorithm searches, thus, for non saturated beams that are neighbours of the intermediary beam. The algorithm then calculates how many packets to move from the first to the intermediary beams and from the intermediary to the third beam. For this, it solves a series of inequalities that involve the number of packets from the two sessions in the three different beams and the transmission rates in each beam. As stated in the outline, the volume of packets to move depends on the sessions' and beams' load.

### Discussion

One key factor considered primordial is saving resources and using them in the most efficient manner. On that account, the allocation should evolve taking into account the impact of the changes in the transmission time. The algorithm firstly orders the possible swaps between beams by the aforementioned ratio between transmission rates. The search follows this order, ensuring the more costly swaps are left for last resort.

This procedure ensures not only that little time is used, but also that a valid configuration is found in the more extreme cases, that is, when the majority of beams have great loads.

The algorithm should find a solution for every case such solution exists, even for extreme cases. In order to assess the fulfilment of this requirement, the algorithm is compared against a linear programming problem solved in a MATLAB environment. The mathematical model is supplied with information about the paths available for each session and the associated traffic. Using this model, a solution is sought for different conditions. From this, we expect that the algorithm reaches a solution for every case one is found for the linear programming problem.

Figure 4.4 compares allocations obtained with the optimization and the load balancing algorithm. The global and each individual beam's transmission time are charted, with successively

## Algorithms

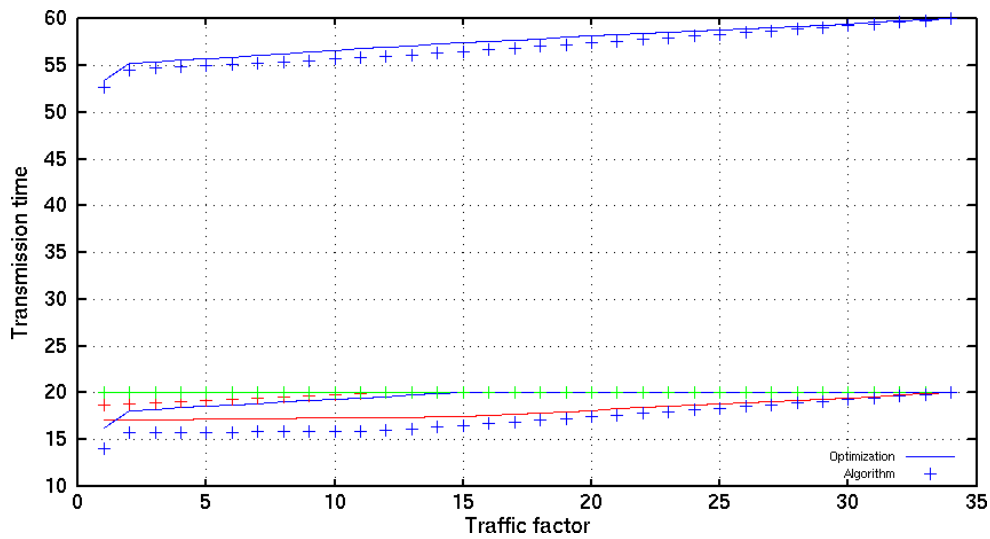


Figure 4.4: Comparison of results from the algorithm and an optimization. Lines represent the optimization.

more packets added to a certain session. As the Figure 4.4 shows, the algorithm is able to attain a solution to every solvable configuration.

## 4.4 Packet discard policy

### 4.4.1 Introduction

The system might not be able to accommodate the requested capacity, given the restrictions upon it. In this situation, a portion of the incoming packets needs to be discarded as the system plainly is not able to cope with the traffic volume. The discarded packets can be retransmitted in the following window, depending on the delay window tolerance. Elapsed that tolerance, the packets are rejected.

Different policies are considered depending on the satellite operator's needs. The discard policy shall be divided in the inter-beam policy and the intra-beam policy. This division is based on the two different levels at which the algorithm works: beam and session levels. We can identify, in each sub-policy, two steps to the packet discard process: a) determining which beams/sessions will undergo delays and b) how many packets to delay in each beam/session.

Given a configuration that the load balancing algorithm is not able to solve, the last iteration's result is observed. Sessions using problematic beams are selected for reducing traffic. The number of packets is slashed in each one according to a specific policy.

In order to reach a solution quickly, the algorithm firstly cuts dramatically the number of packets and then resorts to a method such as the secant method to reach a satisfactory solution. The secant method is a root-finding algorithm that uses successive roots of secant lines to better approximate a root of a function; the algorithm uses a similar method. The first cut determines

roughly the proportion of the cut in each session. The discard policy in place dictates how the algorithm administers the cuts.

### 4.4.1.1 Inter-beam policy

In an invalid configuration, the algorithm has to determine where and how to discard packets. Firstly, it decides in which beams to act, presumably in beams whose transmission time is over the limit. In a system without load balancing, the first step is trivial. As the traffic cannot be moved to neighbour beams, no other option than to intervene in the problematic beams remains. On the other hand, in a system with load balancing, it might be possible to deal with a problematic beam just by discarding packets from neighbouring beams. Whether penalizing a non-saturated beam is acceptable will depend on the policy in place. Nonetheless it is an additional degree of freedom for the algorithm.

The algorithm's performance should be benchmarked against the policy in place. One basic policy is to discard the same number of packets in each beam. Or discard a certain number proportionally to the beams time over the limit. The policy suitability depends on the requirements that the operators have. Depending precisely on those requirements, the algorithm can take into account historical data: average traffic, number of previously discarded packets, high traffic events, etc. These factors are important to maintain certain guarantees or impose limits. For example, if the average traffic for a given beam is above the originally settled, this beam can endure greater delays in detriment of others. On the other hand, if a beam's traffic has been constantly delayed, the algorithm can try to taper the cuts for some time. Thus, gauging traffic patterns on a certain time-frame allows for an enhanced observance of the policies.

In order to speed up the search for a solution, the system could map the valid extreme configurations in a lookup table. Extreme configuration should be interpreted as a configuration where one or more beams are saturated.

We run simulations with an array of varying number of packets in different sessions. In each simulation, the beams have different loads. We end up with a curve as shown in Figure 4.5, plotting the transmission time from beam 2 against that of beam 1. The data is obtained through the solving of the linear programming problem. The maximum transmission time allowed for each beam is delimited by the two lines. The charted data points represent the most extreme cases where no packet discards are necessary. These points are obtained after looking into a wide variety of traffic distribution. As different traffic patterns may lead to the same transmission time at the beam level, it is possible that, for a given time, the configuration is indeed invalid. This happens as the ability to spread traffic to neighbour beams varies depending on the availability of alternate paths.

With the valid configurations mapped, the algorithm can aim at one of the points. Aiming at each of the configurations implies a different behaviour. One can discard more traffic from one beam to spare another, presumably a more important one. The choice depends on the policy in force.

The implemented algorithm does not map valid configurations due to its perceived impracticability. In fact, actual systems may see concurrent interaction between 3 or more beams. Thus,

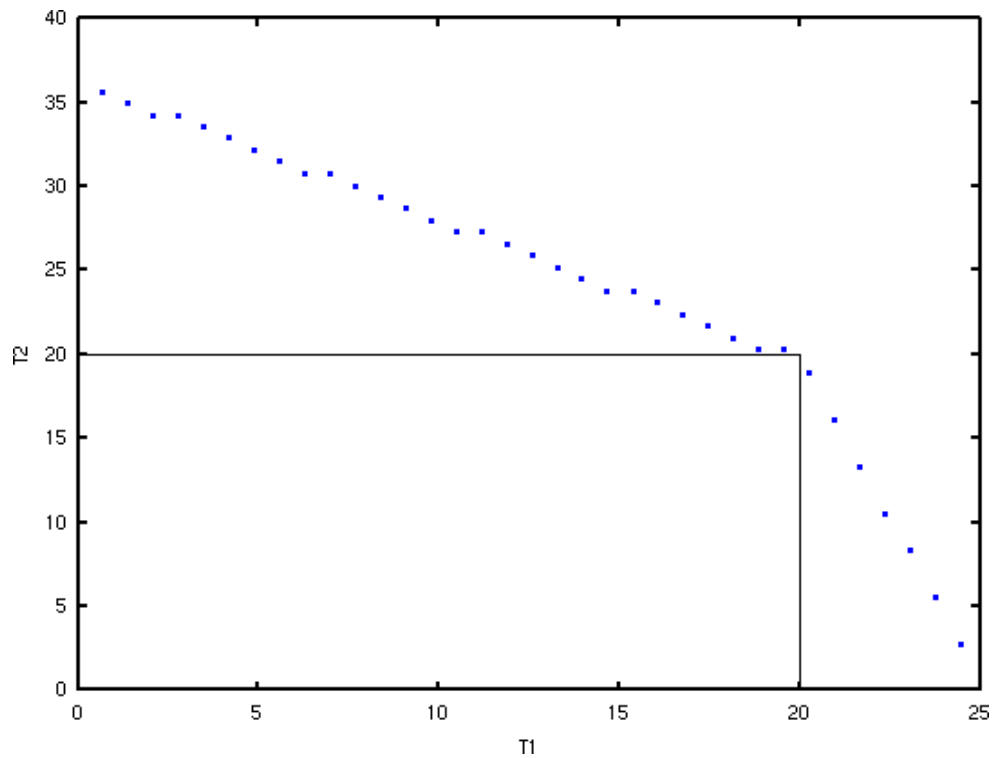


Figure 4.5: Valid configurations in the stable demand region, obtained with the optimization formulation; charted transmission time of neighbour beams.

the mapping would demand a lot of processing power and run the risk of not portraying usefully the system.

#### 4.4.1.2 Intra-beam policy

Once the algorithm decides roughly the ratio of transmission time to slash in each beam, the next step is to figure how to accomplish the cut. Naturally, in order to reduce the transmission time, some sessions have to see a portion of their packets delayed. The algorithm needs to determine firstly which sessions will be considered and what rate to discard. Let us consider the generic algorithm outline:

1. Amidst the problematic beam's sessions, choose the ones that will endure delays;
2. Determine the dimension of the discards, or their proportions in each session.

These steps may be more or less blended depending on the specific policy. Turning to the specifics, one aspect to consider is that sessions have different characteristics. They may be more or less tolerant to transmission delay, depending on the type of content being distributed, or have different priorities.

Considering a policy that is content agnostic and aims at minimizing the total number of packets dropped. With such constraints, the algorithm can discard packets from the session with the

---

**Algorithm 3** Discard algorithm pseudo-code.

---

```

for all Beams  $b$  do
  if  $transmissionTime(b) > maxT(b)$  then
    for Session  $s \in sessions(b)$  do
       $numPackets \leftarrow numPackets + xsl(b,s)$ 
       $sumTNumPackets \leftarrow sumTNumPackets + rsl(b,s) * xsl(b,s)$ 
    end for
     $perc \leftarrow (T(b) - TMax(b)) / nPackets * sumTNumPackets$ 
    for Session  $s \in sessions(b)$  do
       $mssl(s) \leftarrow mssl(s) - mssl(s) * perc$ 
    end for
  end if
end for

```

---

worst transmission rate and proceed to the next session; until the beam's transmission time falls below its limit.

A similar policy is to, like the previous, be content agnostic but aim at inflicting the same cuts in every session. In this way, every session would receive a cut proportional to its transmission time in that beam. This policy is the default one used throughout the simulations. We take this policy as default because we aim at modelling a generic system, where no particular information about the nature of the sessions is available. From that standpoint, this policy is rather neutral as its decisions are based solely on the impact a beam's sessions, as a whole, have on its transmission time.

A policy can take into account the type of session's data and the session's priority. In this way, it can rank sessions according to those properties and the number of packets. This priority policy divides the sessions into priority groups. Each group endures discards in percentages proportional to their priority. Altogether, sessions with higher priority risk smaller delays than other sessions.

Alternatively, the policy does not assign priorities and, so, treats every session equally. But, unlike the default policy, treating equally means "equal opportunity". Therefore, sessions with heavier loads should undergo higher percentage cuts, as they represent the great impact on the beam's transmission time. This weight policy performs the discards proportionally to the weight of a session in a beam, meaning its transmission time in that beam.

# Chapter 5

## Evaluation

This section presents the evaluation of the algorithms developed in this thesis. This assessment is primordial to understand the roles the algorithms play and the benefits they bring in this context. On that note, we present firstly the context, the model used for the simulations. The following section describes the metrics used to evaluate the algorithms' performance. Before the analysis of the results, we introduce the simulator used. Finally, we present the results from simulations in a variety of contexts: the load balancing impact on throughput, different packet discard policies, and redundancy.

### 5.1 Simulation model

Various simulations have been performed in order to assess the benefits of load balancing. To this end, the simulation uses a model of the traffic demand, corresponding to the predicted demands as from the Digital Divide Satellite Offer (DDSO) study.

The system environment in the simulations is the following:

- User bandwidth per beam: 250 MHz
- Capacity requirements: DDSO 2010
- Number of beams: 70
- Total Capacity Request: 17386 Mbps
- Transmission window length: 0.5 sec

In the simulation, the system handles packets 10 Kb in size. The simulation assumes a transmission plan with a pre-defined repetition rate with a window length. In the following tests, this window has 0.5 seconds. During a window, each beam has an opportunity to transmit for a specific time.

### 5.1.1 Clients

The model counts 2680 terrestrial stations each assigned to a given beam. In order to reduce the computational complexity, the beams are considered to be covered by another beam at most. This simplification follows the fact that few sessions are covered by more than two beams, or at least have a reasonable path available through additional beams.

Two different traffic configurations are considered in the simulations: the real capacity requirement and a normalized capacity requirement. The real capacity requirement corresponds to the predicted demands from the DDSO study, as mentioned before. The normalized demand has the same total capacity request but it is distributed uniformly among beams. Thus, beams with more sessions have each session with less traffic.

### 5.1.2 Gateways

As explained in section 2.2.2, sets of grouped beams are controlled each by a gateway. The majority of tests are conducted with that assumption. Further simulations are conducted in which load balancing is allowed among any two beams, irrespective of the gateway. The results are compared to assess the impact of the gateway restrictions.

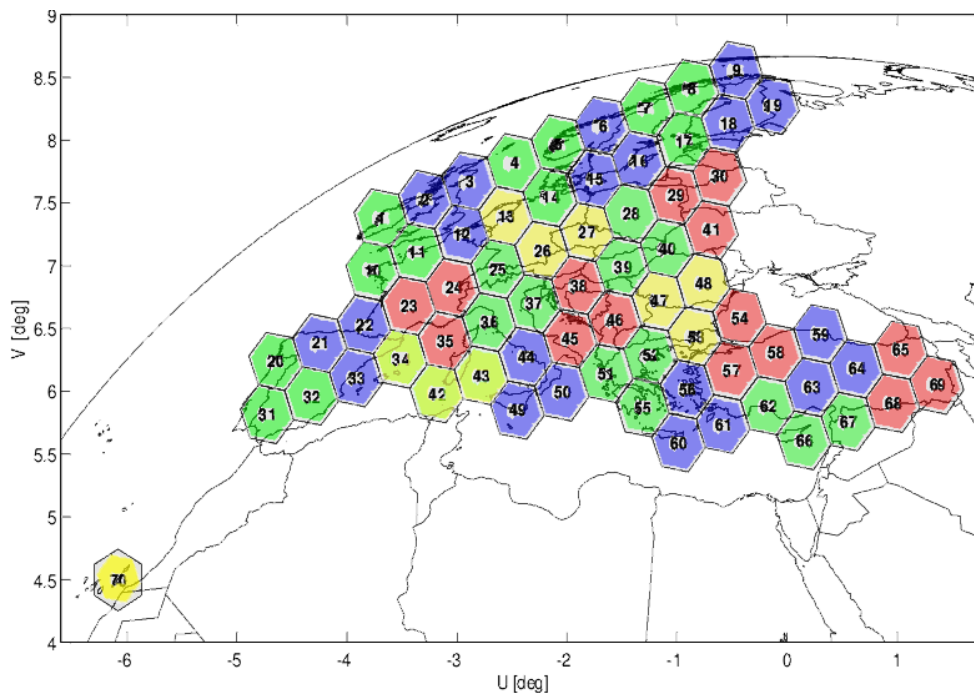


Figure 5.1: Beams grouped by gateway.

Figure 5.1 presents the beam grouping used in the simulations. The beams are all in groups of 3 with the exception of a isolated beam. This configuration mimics the real system thus allowing for reliable results.



## 5.2 Metrics

The results are analysed using different metrics, depending on the type of analysis done. The metrics are detailed in this section. Most of the presented metrics will be analysed as average values, in order to filter noise in the collected data.

### 5.2.1 Transmission time

One straightforward but somewhat crude metric is the beams' transmission time. Each beam has transmission time limited to a determined value and is calculated as previously shown in Section 3.2.1, in Equation 3.1:

$$T^i = \sum_{s_l \in K_i} N_{s_l} / R_{s_l}^i, \quad (5.1)$$

This metric provides a notion of the system's load. The difference between the transmission window and the transmission time depicts unused resources.

Additionally, the global transmission time may be considered:

$$T_{\text{Glo}} = \sum T^i \quad (5.2)$$

The global transmission time is useful to measure the performance of the system, at least when the system is heavily loaded. In this situation, the smaller the difference between the transmission time and transmission windows, the better the performance. This difference is, as mentioned before, indicator of unused resources.

### 5.2.2 Number of transmitted, delayed, and discarded data packets

The plain method to gauge the system's performance is through the analysis of the handling of incoming traffic. That is, to assess the system's traffic capacity based on the portion of packets that are successfully sent, delayed, or discarded.

At the outset of each transmission window, there are new incoming packets to be sent,  $v_{s_l}$ , for each live session. Should there be a delay window for packets, there are, additionally, the preceding delayed packets,  $D_{s_l}$ . Thus, assuming  $N_{s_l}$  to be the number of packets to be send, the subsequent  $N_{s_l}$  is defined as follow:

$$N_{s_l} = v_{s_l} + D_{s_l} \quad (5.3)$$

## Evaluation

After performing the allocation,  $N_{SI}$  can be divided as follow:

$$N_{SI} = \tau_{SI} + D_{SI} + \zeta_{SI} \quad (5.4)$$

Where  $\tau_{SI}$ ,  $D_{SI}$  and  $\zeta_{SI}$  are the number of packets sent, delayed and discarded, respectively.

As with a majority of the metrics, these numbers will be analysed as an average (along) a relatively large number of transmission windows. Thereby, some noise in the data points can be filtered in the resultant data.

### 5.2.3 Load from assigned sessions

In order to assess the actual benefits a beam experiences, it is not exactly the number of packets transmitted by that beam that matters. For example, a beam is saturated when load balancing is performed and when it is not. A hasty conclusion might be that no improvements occurred. Actually, it may be possible that fewer packets, or none at all, were delayed when performing load balancing. In that case, although the beam maintains the same load, there is an increase in offered capacity.

The proposed metric to analyse is, for each beam, the number of packets transmitted for the sessions that are assigned to it, disregarding which path was used:

$$N_{A.Beam} = \sum_{SI \in K} N_{SI} \sum_{s_l \in K; \forall j \in N(i)} \frac{N_{s_l}}{R_{s_l}^i} \quad (5.5)$$

This way, the metric accounts for a beam, traffic from one of its assigned session even if no packet is transmitted through that particular beam.

### 5.2.4 Unmet System Capacity (USC)

One way to translate the system's capacity to handle the requested traffic capacity is the Unmet System Capacity (USC) [ACG<sup>+</sup>10] which is defined by Equation 5.6 where  $C_{rqs}$  is the requested capacity and  $c_{off}$  is the offered capacity.

$$USC = \sum_{i \in N_{Beams}} \max[C_{rqs}(i) - C_{off}(i), 0] \quad (5.6)$$

This metric does not account for the excess capacity offered so  $C_{off}$  can always be assumed to be the actual traffic in each beam. This metric thus measures the system's inability to handle excess traffic and does not provide insight to the remaining unused resources.

## 5.3 Simulation tools

The work involves the development of load balancing algorithms. These must be tested and assessed to verify whether they do indeed perform better than existing alternatives. The simulation is therefore significant in this project as a tool to validate the developed mechanisms.

The tools envisaged to this end are, firstly, MATLAB (MATrix LABoratory), a numerical computing environment and programming language; developed by MathWorks. Afterwards, we develop a custom application in order to evaluate our load balancing algorithms, to test and improve these algorithms.

### 5.3.1 MATLAB

MATLAB is a tool for numerical computing that is widely used in academic and research institutions. It is used for the purpose of providing a simulation environment for the developed algorithms (as used in [PMaSO10]).

MATLAB is used for the initial testing and development of the algorithms. Preliminary simulations are performed firstly on this environment. This tool is used to facilitate the algorithm's initial outline. One specific advantage is the comparison between the algorithm's results and the solution of a linear programming problem, as summed up in Section 4.3. MATLAB allows for quick corrections and feedback. Indeed, the development follows iterative steps as changes are introduced according to the results obtained.

As the algorithms' design matures, we are faced with increasing complexity in the algorithm itself and the underlying simulation infrastructure. Thus, we switch to the development of a custom application.

### 5.3.2 Custom application

A custom application is developed for the simulation of a communication satellite system. It implements a simplified model that encompasses the features relevant to the analysis of the system's performance.

The simulator is used as the mechanism of evaluation of the algorithms. Part of this evaluation will be the benchmarking of the algorithms with different parameters and against existing ones.

One key attribute of the simulator is it being parameterizable. It is developed in the C++ programming language and accepts a variety of parameters. It can perform simulations with different traffic intensities or with or without load balancing. This allows for quickly obtaining data for comparison and analysis.

Furthermore, the developed simulator is able to accommodate the scale of the performed simulations, encompassing nearly 2700 sessions and 70 beams. Being a native application, the execution times — one interesting metric — can be considered reliable to be compared to the times on an actual system.

## 5.4 Load balancing

### 5.4.1 Introduction

This section starts by analysing the general benefits the load balancing algorithm has on the system's effective transmission capacity. The following section presents the effect the delay window tolerance has on the efficiency of load balancing to increase the offered capacity.

As discussed in Section 2.2.2, in actual satellites, there is no single entity controlling all the beams. Rather, beams are handled in small groups by individual gateways.

### 5.4.2 Impact of load balancing on throughput

The major objective of load balancing is increasing the system's throughput, as shall be discussed in this section. This objective is possible by making use of the otherwise unused resources. In this way, less packets have to be delayed or discarded, effectively increasing throughput.

A variety of simulations are conducted in order to assess the improvements in the system's transmission capacity, introduced by performing load balancing. The different scenarios allow for a comprehensive insight into the performance of the algorithm.

Firstly, are presented the results from performing load balancing globally, ignoring the gateways. The next section analyses the impact of load balancing performed among the gateways. We then compare and discuss the various results obtained.

#### 5.4.2.1 Load balancing performed globally

In this situation with load balancing performed globally, the system would be expected to achieve the best throughput increase. Since every alternative path is available, even from beams from different gateways, the system has the greatest flexibility.

Two parallel sets of simulations were run, one without load balancing and the other with it. From the simulations we obtain the load per beam, as mentioned in Section 5.2.3. The difference between the values in the simulation with load balancing and the one without gives the improvement. As with a few of the following figures, three different metrics are charted. The number of beams that saw improvements and the overall improvement among those same beams. The improvement is measured in percentage, relatively to the transmission time without load balancing and in absolute terms, in Megabits per second (Mbps).

Figure 5.2 shows the performance improvement with load balancing performed globally, with normalized demand. The different improvements are registered, as the demand factor is successively increased. This specific configuration, with normalized demand, has the best improvement recorded. A striking result emerging from the data is that, at one point, nearly every beam experiences an improvement, and the overall improvement reaches 1300 Mbps. This may result from the values being averages from 1500 transmission windows. Thus, only a fraction of the beams experiences improvements at any given point. Interestingly, while the global improvement

## Evaluation

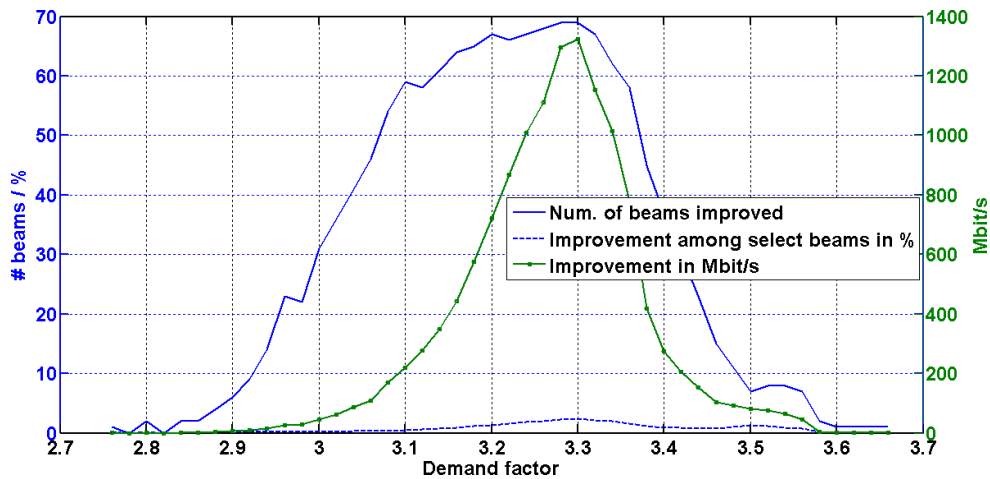


Figure 5.2: Improvement with load balancing vs. without: with load balancing performed globally

is the best, the average improvement per beam is rather small. The fact that so many beams have improvements clearly compensates the small improvement per beam.

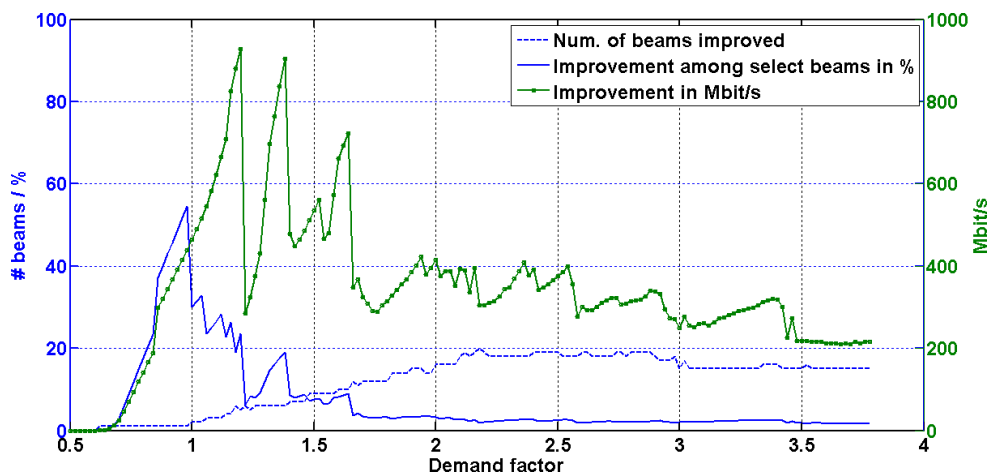


Figure 5.3: Improvement with load balancing vs. without: with load balancing performed globally and real demand

Turning to the outcome with real demand data, the results are notably different. Figure 5.3 depicts the performance improvement measured. Unlike the results with normalized demand, the improvement curve does not have a “bell curve” shape. Moreover, the systems experiences improvements, even with high demands.

In fact, we start to measure improvements even with a diminished demand. This can be explained by the rather large disparity among the beams demand. Heavily loaded beams overload easily even when the system is not experiencing especially high demand. The system is able to move a large portion of traffic owing to the presence of heavily load beams neighbouring sparsely load ones. This predicted high improvement is apparent in Figure 5.3, where an improvement

in the 50% range is measured for a single beam. Between traffic factors of 1 and 1.5, the overall improvement reaches 900 Mbps, a considerable value. As more beams saturate, the average improvement among the beams decreases but is able to maintain a good value in the prospective demand factors.

In larger demand factors, the average improvement continues to be rather remarkable. Interestingly, the absolute improvement decreases smoothly allowing the system to retain improvements even in great load factors. Thus, the algorithm provides improvements even in scenarios of extreme demand.

### 5.4.2.2 Load balancing performed within each gateway

Although theoretically inferior compared with the previous situation, the concrete nature of satellites imposes little or no cooperation between gateways. Ergo, the results in this configuration, within gateways, are the more indicative of the actual impact of load balancing on the system's capacity.

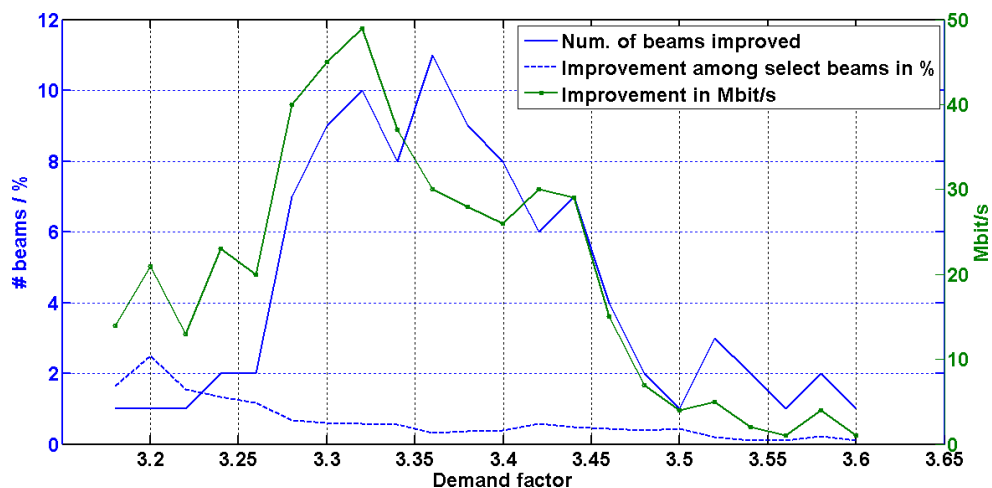


Figure 5.4: Improvement with load balancing vs. without: with load balancing performed within gateways

Analogously to the previous figures, Figure 5.4 shows the measured improvements performing load balancing within each gateway and with uniform demand. This configuration sees the worst improvements, likely due to conjunction of the uniform demand and the restrictions imposed on the load balancing algorithm by the gateways. A small number of beams, that experience higher throughput than the average, translates only in a marginal percentage improvement. Still, this small number, that peaks at around ten beams, represent roughly one tenth to one seventh of all the beams.

As stated, this configuration depicts the smaller improvements, due to setting factors. As such, this improvements can be understood as possibly the worst-case scenario in terms of the presence of factors the algorithm takes advantage of. On that account, we find that the algorithm is still able to improve throughput in adverse conditions.

## Evaluation

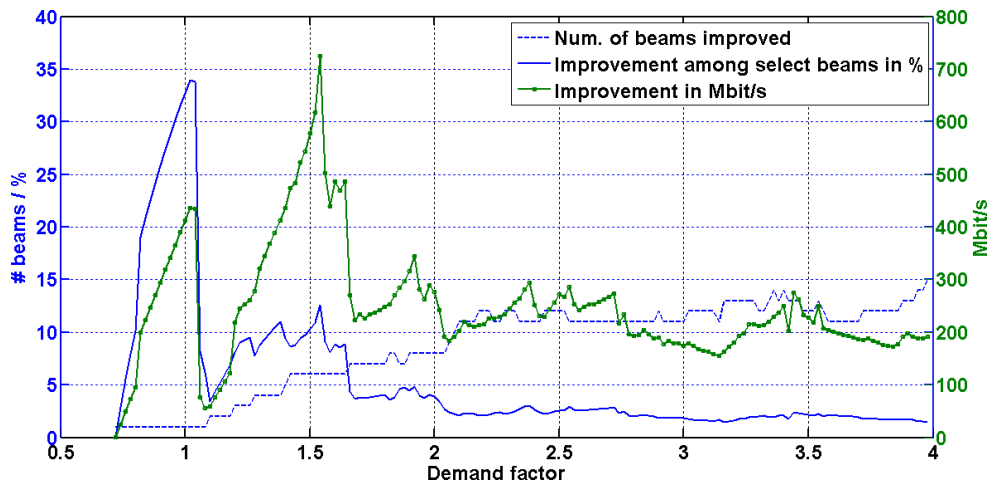


Figure 5.5: Improvement with load balancing vs. without: with load balancing performed within gateways and real demand

Figure 5.5 shows the improvement with real demand data. The number of improved beams rises steadily as the demand increases, between demand factors 1 to 2.5. The absolute improvement rises quickly at the beginning and then stabilises at around 200 Mbps. At the peak of the registered improvement, in a plausible demand factor, the system sees an improvement of around 700 Mbps, which represents around 4% of the base demand.

### 5.4.2.3 Comparison between load balancing performed global and within gateways

We have previously presented the results of load balancing in these two conditions. It is now important to confront the two sets of results. A comparison of the two configurations reveals that load balancing performed globally yields mostly better improvements, as would be expected since the system faces less restrictions.

Figure 5.6 compares the improvements in percentage and the number of beams improved, when load balancing is performed globally and within gateways. In the lower traffic demand range (factors 0.5 to 1.6), the average improvement per beam tends to be higher in the global case. With higher traffic demand, the average improvement per beam is approximately the same, even if, between 2.0 and 3.0 demand factors, the global improvement is just about double of that of within gateways.

Figure 5.7 compares the global improvements in capacity with and without load balancing, showing the gains in throughput against a simulation without load balancing. Interestingly, the two curves tend to follow the same pattern, maintaining mostly only a rather small difference between them. Greater throughput improvements are indeed found in the global case.

The comparison between load balancing performed within gateways versus globally is relevant since the global case does not occur in actual systems but yields the best improvements. As seen from these comparisons, the difference is quite acceptable. In fact, lower improvements can be

## Evaluation

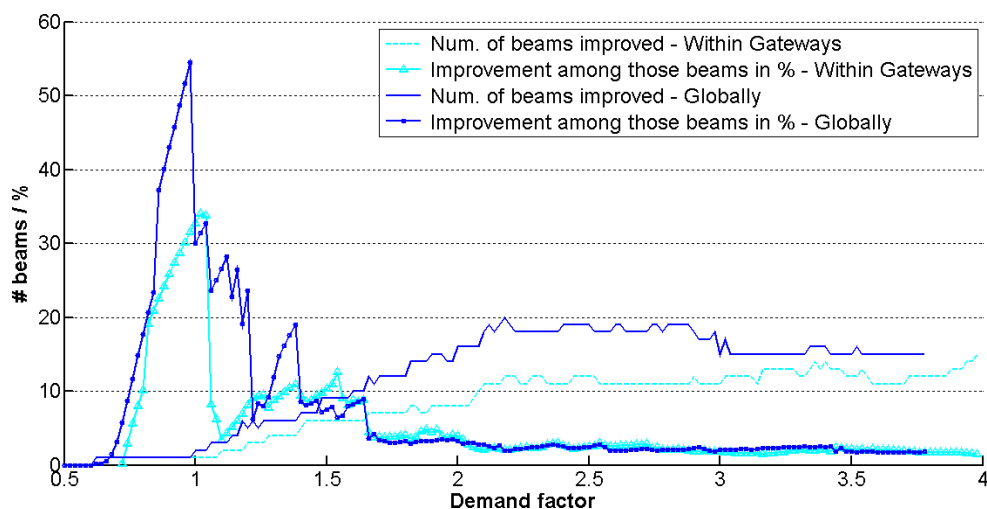


Figure 5.6: Comparison of improvements in percentage and number of beams: globally vs. within gateway.

seen as a trade-off for using current systems without modification and reduced computational complexity as less alternative paths are available.

Moreover, we should note that the system experiences great improvements with average demand and, even, with higher and more unlikely demands. This shows just how much current systems' performance can be improved with this algorithm by making use of unused resources.

### 5.4.3 Delay window effect on load balancing performance

The results previously discussed come from a simulation with a delay window of 4 transmission windows. It was presumed that the delay window has an effect on the impact of performing load balancing.

For the purpose of verifying whether such effect exists, a set of simulations are conducted. The traffic intensity is kept constant but the size of the delay window tolerance is sequentially increased. The results obtained can be compared in Figure 5.8. The chart presents the improvement in the number of transmitted packets: with load balancing compared to without. The values are in percentage and reflect the difference in the average number of packets transmitted by the beam number 50. In fact, a group of simulations was done for each delay window size, with a range of traffic factors — from 3.22 to 3.34. This was done in order to maintain similar load conditions between simulations, as a large delay window may imply a bigger load on the system. Thus, the aforementioned average is the average of the average of the simulation run over 1500 transmission windows.

From the data in the previous Figure 5.8, it is apparent that performing load balancing has a much greater impact with smaller delay windows. From the improvement of 120% with no delay tolerance, the system only sees an improvement of over 30% with 7 windows of delay tolerance, or



## Evaluation

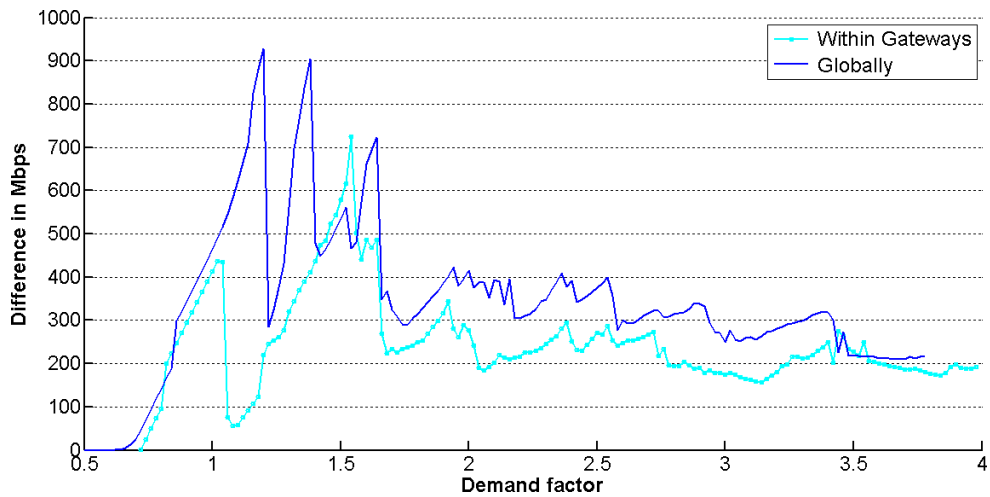


Figure 5.7: Comparison of improvements in Mbps: globally vs. within gateway.

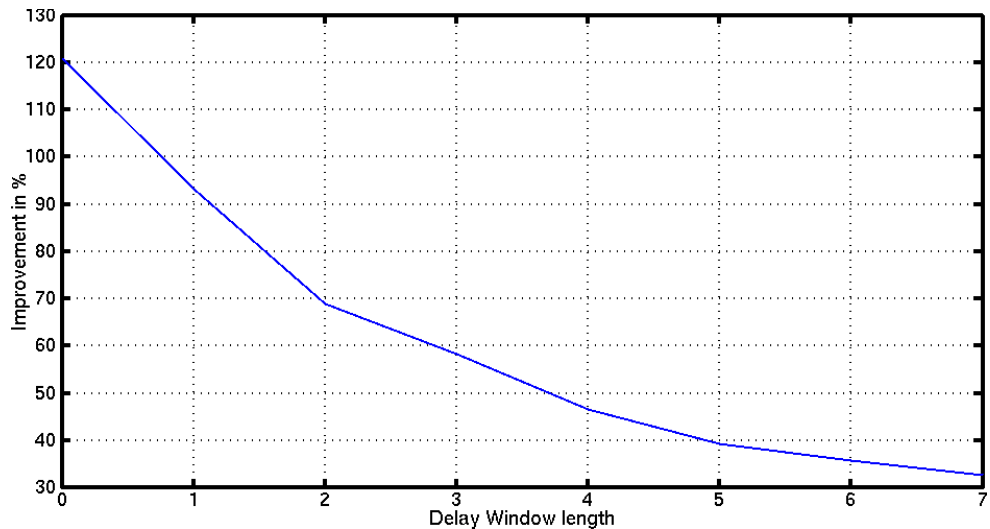


Figure 5.8: Improvements in the number of discarded packets, in the cases of different number of windows of delay tolerance.

about one fourth of the former value. Indeed, the bigger the delay window, the better the chances of eventually transmitting a delayed packet, without discarding it, at a later transmission window.

We can conclude that, while load balancing offers benefits with larger delay windows, the improvements are truly attenuated. Nevertheless, we should note that not every application admits long delays. Thus, load balancing stands as a better solution against increasing the delay window.

### 5.4.4 Computational complexity

An important characteristic we cannot overlook in the design of the algorithms is their computational complexity. An algorithm that finds the optimal solution but runs in an unreasonable amount of time is useless. In our case, we are bound to the length of a transmission window in the satellite.

## Evaluation

It is assumed that new incoming packets are received at the beginning of a transmission window, so the algorithm has a time frame equal to the size of that transmission window to allocate traffic to the next window. This conceded time frame comes off as relatively ample for the type of complexity presented by the algorithm.

Turning now to tests conducted to assess the computational time our algorithms require. Two sets of simulations are run: the first with normalized demand and the second with real demand data, both as described in 5.1.1. The objective of these two sets is to assess the algorithm's performance with different portions of available alternative paths and demand patterns. In fact, with evenly loaded beams, we deem the algorithm sweeps a higher number of beams looking for possible swaps. This is due to the larger number of problematic beams and the reduced room for manoeuvre in the remaining beams.

Each set is made up of three simulations, the first one without load balancing to have a base of comparison. Two other simulations are run: one with load balancing performed within each gateway and the other with load balancing performed globally. We expect the latter to run to require more computational power as a wider array of alternative paths are available to explore. The simulations are run with no delay window to have the same incoming traffic in all.

A simulation is run for 500 transmission windows in order to filter small discrepancies and have a more easily comparable value. The obtained time includes not only the algorithm's computations but also additional procedures necessary for the maintenance of the simulated environment. For this reason, the execution time without load balancing is higher than zero.

On the technical part, the simulations performed in this thesis are run on a 2011 average consumer desktop computer. Naturally, execution times may not match those on actual satellite systems as the hardware on which the algorithms run is fundamentally different.

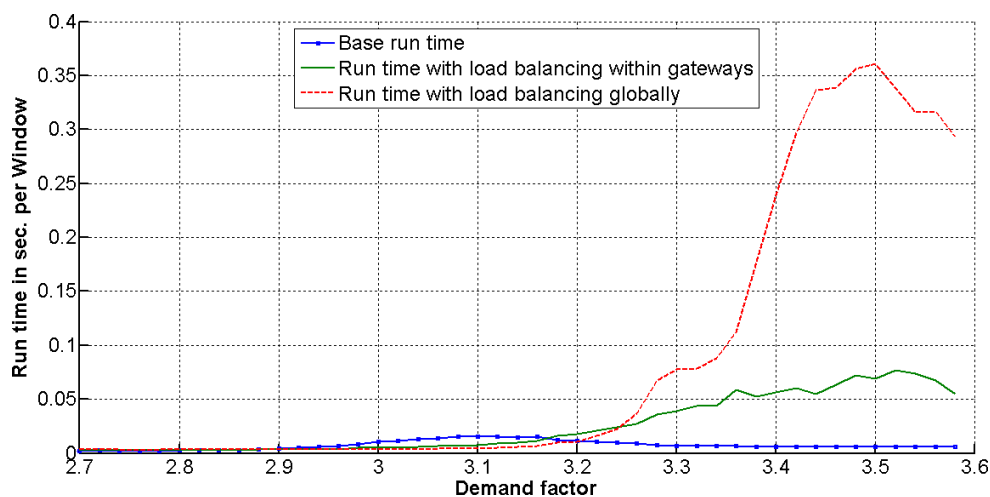


Figure 5.9: Comparison of run time for simulations, with normalized demand.

Turning now to the results, Figure 5.9 compares the execution times of the three simulations, with normalized demand. The execution time with load balancing within gateways peaks at 10 times that of without. Furthermore, the time values show a stable trend with increasing demand.

## Evaluation

With load balancing performed globally, the execution time upholds at reasonable levels up until a demand factor of 3.3. After that point, the time skyrockets to much higher values. Nevertheless, even at its pinnacle value, it is still below the transmission window time for these simulations — 0.5 seconds.

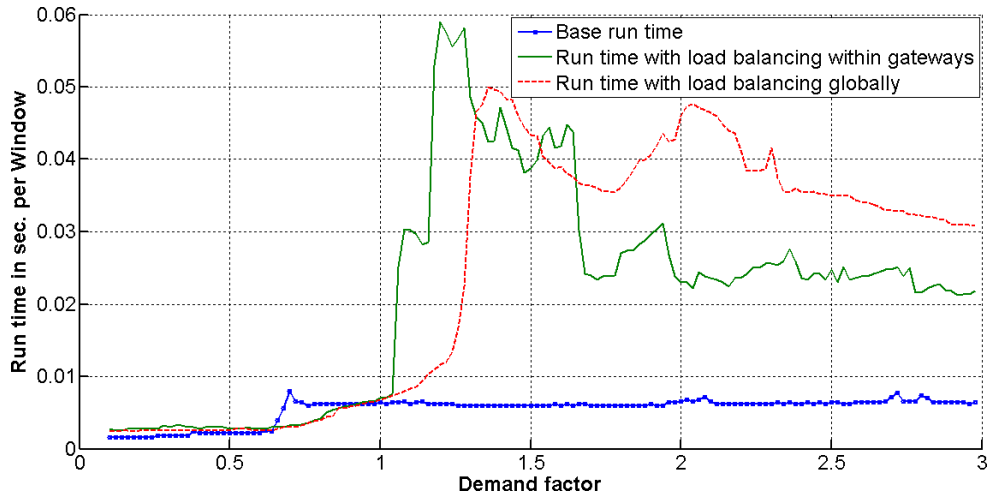


Figure 5.10: Comparison of run time for simulations with real demand.

Figure 5.10 compares the execution times, with real demand data. Like with the previous set, the execution time with load balancing globally peaks at nearly 10 times that of without, but at a lower demand factor as that is region where more improvements occur.

### 5.4.5 Discussion

As is the objective of this study, load balancing proves to be an effective tool for improving the system's throughput. Key factor that make this possible is the use of previously unused resources and flexible coding techniques that allow for a efficient use of the multiple routes.

In a configuration with uniform demand among beams, smaller improvements are measured. In fact, in this configuration the system's resources tend to be exploited evenly, so the beams all saturate at roughly the same point. Therefore, little room is left for further improvements, when it is most needed. Thus, the results with uniform demand must be interpreted with caution because the aim load balancing is exactly to take advantage of unused resources. In this situation, it still does and in proportion to the unused resources left.

On the technical part, as mentioned before, the execution times reported here may not match exactly those that would be expected on actual systems. Thus, these results need to be interpreted with caution. Besides, we concede that the application code could be further optimized and, hence, allowing for better times. Moreover, satellite systems might offload a portion of the processing to each beam, reducing the burden of on-board systems. In this manner, execution time is also minimized as processing occurs in parallel.

The results from these studies are only possible due to the use of flexible coding techniques. In fact, when performing load balancing, a portion of a given session's traffic is moved to another path. As opposed to the classical soft-handover mechanisms, each route is not required to send all the data packets. It also exempts from transmitting a portion of the packets through the two routes. Instead, the system is able to effectively reduce the load on a beam and increase only marginally on another, as minimally as possible.

## 5.5 Fair Access Policies

### 5.5.1 Introduction

The Fair Access Policies determine the algorithm's behaviour towards discarding packets. The default policy employed in much of this work has the objective of maximizing the global throughput. To this end, the algorithm searches the beams that have to endure cuts and then does the cuts in the sessions whose traffic is transmitted by the problematic beam. Every session undergoes a cut that is proportional to its transmission time, thus ensuring that the impact is directly proportional to the weight each session has on a given beam.

A policy's behaviour can deviate from the global throughput maximization as stated above. This section presents two alternative policies: the first dealing with sessions with different priorities, and the second uses the weight of a session in a beam as penalisation factor.

### 5.5.2 Results

A likely scenario in a communication satellite is that some sessions have a higher priority when routing traffic. Such sessions should receive preferential treatment in case of high load. This translates to shifting some of the necessary delays to non-priority sessions.

Figure 5.11 presents the divergence between the number of packets sent by beam 13 with and without the priority policy. The figure's chart shows the difference in the number of packets sent with the priority policy in place, compared to without. The values are in percentage relatively to the beam's traffic with the priority policy.

From this figure, it is apparent that the algorithm is indeed performing its task as there is a considerable improvement in the number of transmitted packets from priority sessions. As the load increases, a portion of the beam's traffic is delayed and this policy favours the delay of the other sessions. Thus, less packets from those sessions are sent, as is depicted in the chart.

This policy shifts the objective from maximizing global throughput to benefiting priority sessions. As would be expected, throughput is thus affected. The observed decrease in throughput can be attributed to the objective shift.

Turning now to a policy that assigns no priorities. Admitting that every session is equally important, it arises that sessions experiencing heavier loads should not impair the remaining ones. We can outline a policy that cuts traffic proportionally to the demand experienced.

## Evaluation

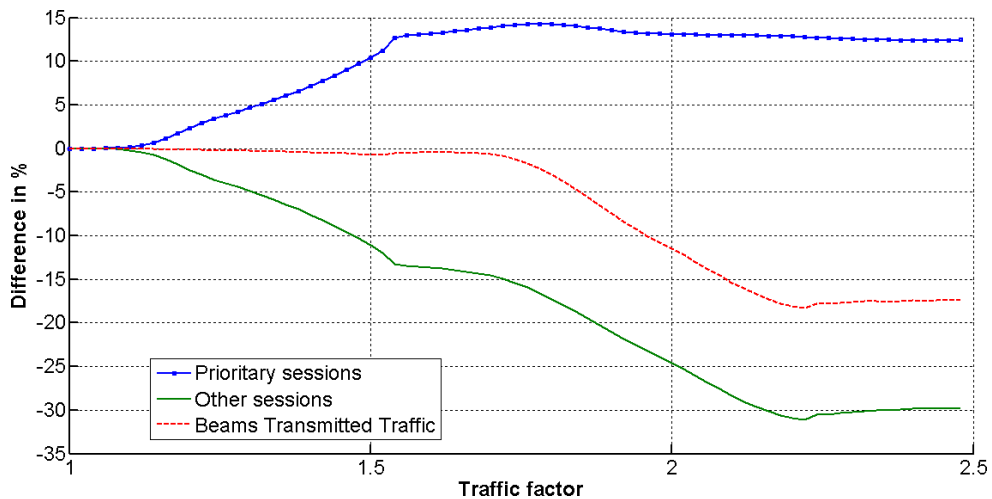


Figure 5.11: Comparison between the default and priority policies: traffic in beam 13 and traffic in the two groups of sessions discriminate. Comparison between the default and priority policies: traffic in beam 13 and traffic in the two groups of sessions discriminated.

As discussed in Chapter 4, this policy discards packets in a beam like the default policy, in average. Instead of performing the same percentage cut in every session, the policy performs a cut proportional to the weight of the session in that beam, in terms of transmission time.

Two distinct sets of simulations were run, one with the default policy and the other with the weight policy. As a means to assess the impact of the new policy, some sessions are deployed with more or less demand than the remainder of their beam's sessions. So, roughly one tenth of the sessions has twice the demand in their beam and another one tenth has half the demand.

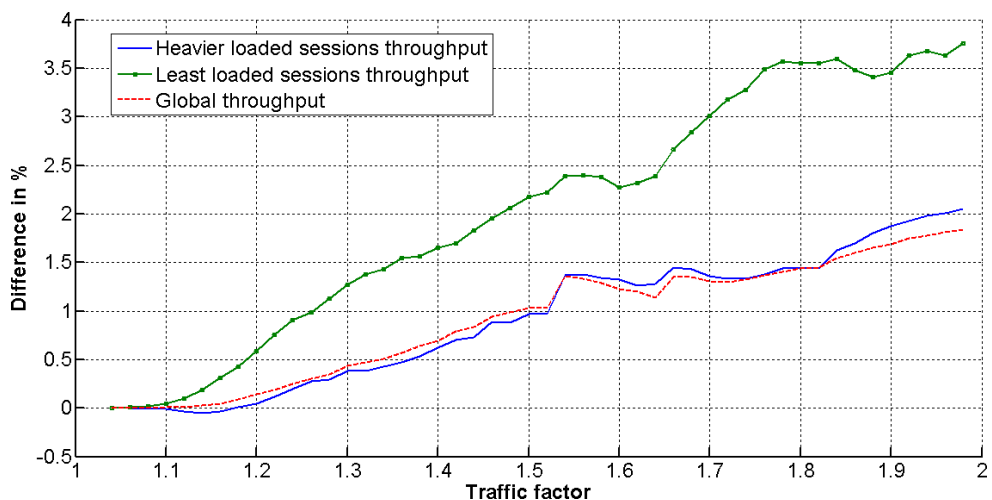


Figure 5.12: Difference in throughput compared to the default policy: among heavier loaded sessions, less loaded sessions and globally.

Figure 5.12 shows the impact this policy has on throughput, from the data collected from the aforementioned simulations. Interestingly, the heavier loaded sessions experience a throughput

increase with the new policy but so does the global throughput. There is a significant correlation between these two improvements.

The improvement in the least loaded sessions is notably higher than that of the global throughput. As the overall demand increases and, therefore, the proportion of discarded traffic too, the improvement increase in these sessions outpaces the global improvement increase, demonstrating the policy in force.

### 5.5.3 Discussion

Fair Access Policies define the behaviour of the resource allocation algorithm. The default policy used throughout this work aims at maximizing the global throughput and, thus, can be reasonably measured. Alternative policies may have more subjective objectives, hence blurring the evaluation of their effectiveness. Effectively, the success assessment depends on the operator's needs, each policy being judge based on different criteria.

The policies can act at the beam or session levels. Their reach depends on the intend.

The results of the weight-based policy study show that sessions with the less demand experience the greater improvement. One unanticipated finding is that there is an improvement in the global throughput. This finding suggests that the default policy is not close to optimal or, likely, that the iteration process the algorithm follows to find a solution could still need some further fine-tune.

Further studies should be undertaken to evaluate the impact of this type of policies. One aspect that is not explored is the use of demand averages spanning across several transmission windows. Using such averages would allow for fairer decisions by the algorithm, such as not penalizing sessions with demand spikes but rather sessions with steadily above average demands.

## 5.6 Redundancy and network coding

### 5.6.1 Introduction

Wireless communication channels are prone to packet losses. Satellite communication is particularly concerned as relative long latency are found. Thus, the system takes more time to notice missing packets and there is a significant delay in the retransmission of the packets.

We resort to network coding for providing redundancy to the satellite transmissions. Redundancy has the negative outcome of decreasing the actual system's capacity. In fact, more data needs to be transmitted than the number of information packets.

The simulations performed focus on the impact of two different types of coding techniques: Dense Network Coding (DNC) and Systematic Network Coding (SNC). In DNC, for a batch of  $N$  packets, the system sends  $N + M$  coded packets, each being a linear combination of the original packets. In order to recover the original packets, the terminal must receive at least  $N$  coded packets. If less than  $N$  packets are received, the terminal is not able to recover a single packet. In SNC, for a batch of  $N$  packets, the system sends  $N$  packets, copies of the originals, followed by  $M$  packets,

linear combinations of the originals. In this case, the terminal can decode any packet whose copy it receives. Additionally, it can recover a lost data packet if it receives at least  $N$  packets out of the remaining  $N + M - 1$  packets.

## 5.6.2 Results

In order to assess the overall impact of both the benefits and downsides of redundancy, we conduct various sets of simulations with different erasure probabilities. In every simulations, the transmitted packets are prone to erasure according to a specific erasure probability. These probabilities range from a relative low value to higher ones: 15%, 30%, and 50%.

In each set, a simulation is run without any redundancy to obtain a base throughput and the number of packets lost. The two other simulations are run employing different types of redundancy, specifically: Dense Network Coding (DNC) and Systematic Network Coding (SNC).

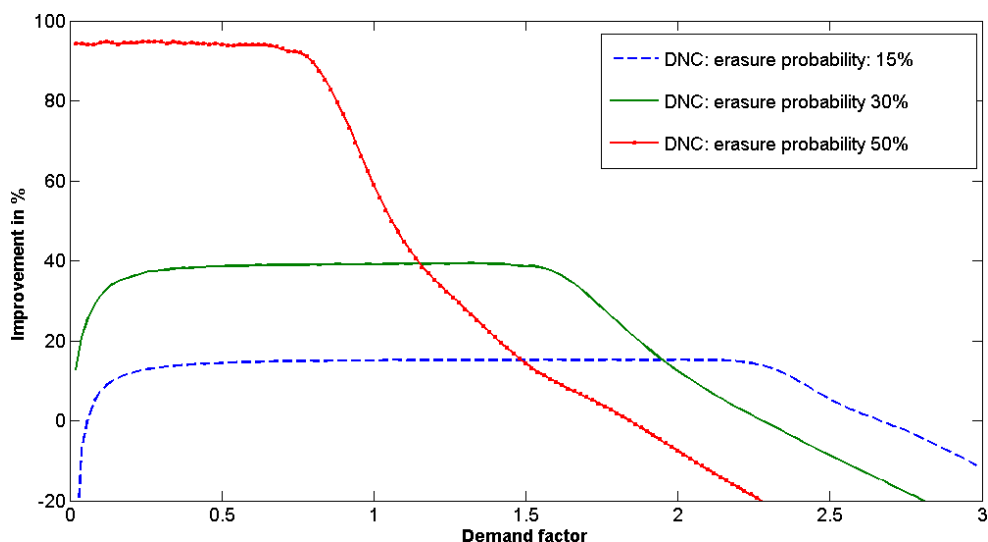


Figure 5.13: Improvement in successfully transmitted traffic with DNC, with different erasure probabilities.

Figure 5.13 compares the improvements with DNC, with the different erasure probabilities. The chart plots the percentage improvement of successfully transmitted packets in each case, compared to the results without redundancy. With an erasure probability of 50%, DNC brings improvements that top nearly 95%. In this situation, without redundancy, naturally half the packets are lost. When we introduce redundancy, as close to every packet is successfully transmitted, this number is doubled resulting in a nearly 100% improvement. With lower erasure probabilities, the maximum improvement is naturally smaller. As with the previous case, the improvements with 15% and 30% erasure probability are close to the best scenario.

Figure 5.14 compares the improvements, this time, with SNC, with different erasure probabilities. The chart is analogous to the previous one. It is apparent that SNC yields, generally, slightly

## Evaluation

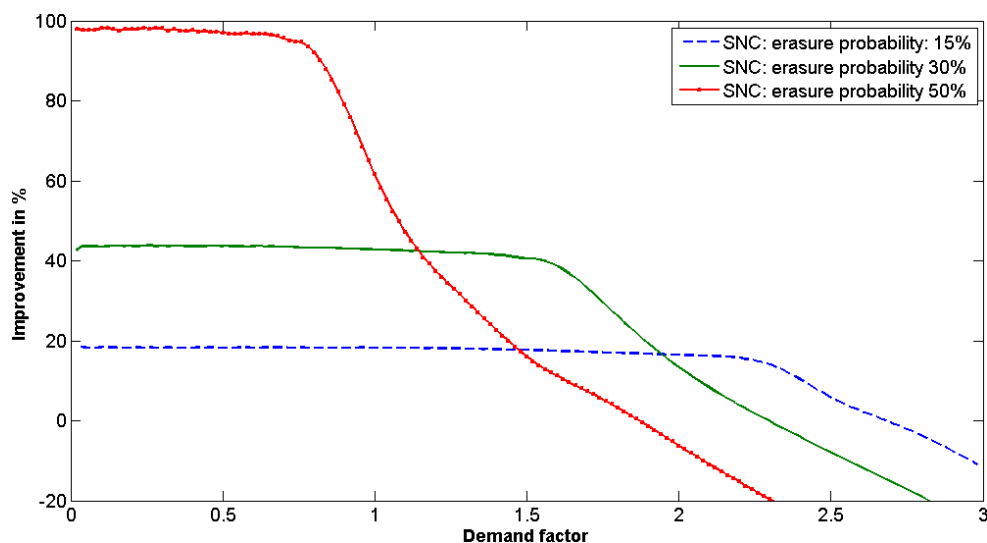


Figure 5.14: Improvement in successfully transmitted traffic with SNC, with different erasure probabilities.

better results than DNC. With an erasure probability of 50%, the improvements reach the 98% mark.

Both methodologies exhibit good performance, with SNC having values usually higher by 3–4 percentage points. As demand increases, the system has less and less unused resources. As a consequence, the overhead that accompanies redundancy permits less data to be accommodated and, thus, the performance improvements plummet. With higher erasure probabilities, more redundancy is needed so improvements start to diminish sooner, as traffic increase. Nevertheless, redundancy is beneficial up until a traffic factor of 2, with a 50% erasure probability, and higher demand for smaller erasure probabilities.

### 5.6.3 Discussion

These findings suggest that network coding applied to redundancy is beneficial to the system overall performance. Redundancy is particularly convenient for delay-constrained application, where avoiding packet loss prevents delays which would eventually lead to discards due to the time constraint.

The results of this study support the idea that while redundancy does indeed offer resiliency to the system, it does so up to a point. The overhead that accompanies redundancy eventually neutralizes its benefits and even worsens the system's transmission capacity. Nevertheless, the system sees improvements in throughput, more or less significant, up to a relatively high demand. These methodologies prove, thus, to be an important tool in guaranteeing a system's high throughput.

In order to have the best throughput with the possibility of packet erasures, we suggest that redundancy should be only exploited in periods when the system is experiencing relatively low demand. Beyond a determined point, the system should quit or reduce redundancy so that it can



## Evaluation

produce better throughputs. This point depends on the concrete nature of the system. Moreover, the selection of the type of redundancy should be made according to the characteristics of the system.

## Evaluation

# Chapter 6

## Conclusions

### 6.1 Conclusions and future work

This thesis sets out to study the impact that load balancing has on a multi-beam satellite's throughput. Owing to the costly nature of deploying such systems, even small performance improvements are very relevant. Much research is dedicated to this topic, driven by telecommunication operators. This project seeks a performance improvement by taking advantage of alternative paths offered by neighbour beams. Jointly, it aims at assessing the role that network coding has in this context.

We studied the context of the problem and related technologies. Such technologies, in the scope of multi-beam satellites, include beam-hopping and power allocation schemes. Furthermore, we analysed the application of techniques that constitute a novelty in this context. We examined multipath routing as we intend on using alternative paths concurrently with the assigned route. Related are network coding techniques that allow for redundancy in the system with minimal overhead.

The core of this thesis lies in the design and implementation of algorithms for load balancing and resource allocation. Additionally, in order to evaluate the system's capacity improvements, a custom simulator is also developed. The simulator allows to test the algorithms with diverse traffic demands and other environment parameters.

The results of this investigation show that performing load balancing does bring improvements to the system's throughput. These improvements can be typically in the 10% range among the beams that experience better performance. Unsurprisingly, without the gateway restrictions greater performance improvements. However, the difference in improvements between the two cases, while existent, are relatively small. We consider the results with restrictions acceptable and satisfying our objectives. Nevertheless, we foresee an extension of this work exploring communication between gateways as a way to perform load balancing across beams in different gateways.

Furthermore, the algorithm reaches good improvements, performing computations in an acceptable timeframe. A satellite's transmission window allows for a relative long computational time. Yet, the algorithm is shown to have a small execution time allowing for extra complexity.

## Conclusions

Flexible coding techniques play an important role in the exploitation of multiple paths. Moreover, they provide a transparent solution for providing reliability of the communication channel. These two roles are noteworthy and important for the deployment of our load balancing algorithm. The redundancy benefits are shown to be substantial, allowing for low packet erasure in an unreliable channel. Moreover, the improvements persist, albeit decreasing, with high demand factors.

Concluding, it is our belief that the objectives of this thesis have been achieved. The applicability of the developed algorithms and findings in real-world commercial satellites makes this thesis an interesting endeavour and a positive contribution to this ever evolving area.

# Appendix A

# Appendix A

Table A.1 shows the demand in Mbps for each beam, as from the DDSO study.

Beam	Demand (Mbps)	Beam	Demand (Mbps)	Beam	Demand (Mbps)
1	192.352941	25	655	49	173.117647
2	273.568627	26	691	50	36.3333333
3	290.666667	27	746	51	365.470588
4	1	28	213.72549	52	599
5	76.9411765	29	599	53	252.196078
6	115.411765	30	136.784314	54	17.0980392
7	36.3333333	31	545	55	365.470588
8	36.3333333	32	429.588235	56	115.411765
9	1	33	115.411765	57	192.352941
10	136.784314	34	76.9411765	58	115.411765
11	380.431373	35	365.470588	59	76.9411765
12	453.098039	36	380.431373	60	55.5686275
13	510.803922	37	399.666667	61	96.1764706
14	1180	38	230.823529	62	153.882353
15	655	39	307.764706	63	173.117647
16	96.1764706	40	136.784314	64	192.352941
17	55.5686275	41	1	65	96.1764706
18	115.411765	42	17.0980392	66	17.0980392
19	17.0980392	43	273.568627	67	36.3333333
20	489.431373	44	438.137255	68	136.784314
21	329.137255	45	329.137255	69	153.882353
22	213.72549	46	153.882353	70	17.0980392
23	346.235294	47	153.882353		
24	399.666667	48	153.882353		

Table A.1: Beam Demand according to the DDSO study

## Appendix A

# References

- [ACG<sup>+</sup>10] J. Anzalchi, A. Couchman, P. Gabellini, G. Gallinaro, L. D’Agristina, N. Alagha, and P. Angeletti. Beam hopping in multi-beam broadband satellite systems: System simulation and performance comparison with non-hopped systems. In *Advanced satellite multimedia systems conference (asma) and the 11th signal processing for space communications workshop (spsc), 2010 5th*, pages 248–255, sept. 2010.
- [ACLY00] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, jul 2000.
- [CC05] J.P. Choi and V.W.S. Chan. Optimum power and beam allocation based on traffic demands and channel conditions over satellite downlinks. *Wireless Communications, IEEE Transactions on*, 4(6):2983–2993, nov. 2005.
- [CRS99] Israel Cidon, Raphael Rom, and Yuval Shavitt. Analysis of multi-path routing. *IEEE/ACM Trans. Netw.*, 7:885–896, December 1999.
- [DP11] A. Destounis and A.D. Panagopoulos. Dynamic power allocation for broadband multi-beam satellite communication networks. *Communications Letters, IEEE*, 15(4):380–382, april 2011.
- [FLBW06] Christina Fragouli, Jean-Yves Le Boudec, and Jörg Widmer. Network coding: an instant primer. *SIGCOMM Comput. Commun. Rev.*, 36:63–68, January 2006.
- [HMK<sup>+</sup>06] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, Jun Shi, and B. Leong. A random linear network coding approach to multicast. *Information Theory, IEEE Transactions on*, 52(10):4413–4430, oct. 2006.
- [HR08] Jiayue He and J. Rexford. Toward internet-wide multipath routing. *Network, IEEE*, 22(2):16–21, march-april 2008.
- [HSC<sup>+</sup>08] Yang Hong, A. Srinivasan, B. Cheng, L. Hartman, and P. Andreadis. Optimal power allocation for multiple beam satellite systems. In *Radio and Wireless Symposium, 2008 IEEE*, pages 823–826, jan. 2008.
- [HZL<sup>+</sup>08] Song Han, Zifei Zhong, Hongxing Li, Guihai Chen, E. Chan, and A.K. Mok. Coding-aware multi-path routing in multi-hop wireless networks. In *Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International*, pages 93–100, dec. 2008.
- [KMM03] Ralf Koetter, Muriel Médard, and Senior Member. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11:782–795, 2003.

## REFERENCES

- [LBAMG07] K.P. Liolis, A. Bolea-Alamanac, C. Morlet, and A. Ginesi. Applicability of fade mitigation techniques to mobile dvb-s2/rcs satellite systems: Accent on railway scenario. In *Satellite and Space Communications, 2007. IWSSC '07. International Workshop on*, pages 1 –5, sept. 2007.
- [LMaS10] D.E. Lucani, M. Mé anddard, and M. Stojanovic. Systematic network coding for time-division duplexing. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 2403 –2407, june 2010.
- [LYC03] S.-Y.R. Li, R.W. Yeung, and Ning Cai. Linear network coding. *Information Theory, IEEE Transactions on*, 49(2):371 –381, feb. 2003.
- [ML12] André Moreira and Daniel E. Lucani. On Coding for Asymmetric Wireless Interfaces. page 5, 2012.
- [MPC08] P. Merindol, J.-J. Pansiot, and S. Cateloin. Improving load balancing with multipath routing. In *Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on*, pages 1 –8, aug. 2008.
- [PMaSO10] A. ParandehGheibi, M. Me anddard, S. Shakkottai, and A. Ozdaglar. Avoiding interruptions - qoe trade-offs in block-coded streaming media applications. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1778 –1782, june 2010.
- [PR11] R. Prior and A. Rodrigues. Systematic network coding for packet loss concealment in broadcast distribution. In *Information Networking (ICOIN), 2011 International Conference on*, pages 245 –250, jan. 2011.
- [RDG04] Rita Rinaldo and Riccardo De Gaudenzi. Capacity analysis and system optimization for the reverse link of multi-beam satellite broadband systems exploiting adaptive coding and modulation. *International Journal of Satellite Communications and Networking*, 22(4):425–448, 2004.
- [RGKR08] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez. An optimization framework for opportunistic multipath routing in wireless mesh networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 2252 –2260, april 2008.
- [SB04] M. Schubert and H. Boche. Solution of the multiuser downlink beamforming problem with individual sinr constraints. *Vehicular Technology, IEEE Transactions on*, 53(1):18 – 28, jan. 2004.
- [SBJ<sup>+</sup>11] B. Shrader, A. Babikyan, N.M. Jones, T.H. Shake, and A.P. Worthen. Rate control for network-coded multipath relaying with time-varying connectivity. *Selected Areas in Communications, IEEE Journal on*, 29(5):1106 –1117, may 2011.
- [VB09] F. Vieira and J. Barros. Network coding multicast in satellite networks. In *Next Generation Internet Networks, 2009. NGI '09*, pages 1 –6, july 2009.
- [VCL08] F. Vieira, M.A.V. Castro, and Jiang Lei. Datacast transmission architecture for dvb-s2 systems in railway scenarios. In *Signal Processing for Space Communications, 2008. SPSC 2008. 10th International Workshop on*, pages 1 –5, oct. 2008.



## REFERENCES

- [YY08] Yamin Sun Jingyu Yang Yuwang Yang, Chunshan Zhong. Energy efficient reliable multi-path routing using network coding for sensor network. In *Computer Science and Network Security, IJCSNS International Journal of*, volume 8, pages 329–338, december 2008.
- [ZL08] Xinyu Zhang and Baochun Li. Dice: a game theoretic framework for wireless multipath network coding. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '08*, pages 293–302, New York, NY, USA, 2008. ACM.
- [ZL09] Xinyu Zhang and Baochun Li. Optimized multipath network coding in lossy wireless networks. *Selected Areas in Communications, IEEE Journal on*, 27(5):622–634, june 2009.