MASTER'S THESIS

Resource Allocation in Ka-band Satellite Systems

by Youyu Feng
Advisor: John S. Baras

CSHCN MS 2001-1
(ISR MS 2001-2)

ABSTRACT

Title of Thesis: RESOURCE ALLOCATION IN KA-BAND SATELLITE SYSTEMS

Degree candidate: Youyu Feng

Degree and year: Master of Science, 2001

Thesis directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

The Ka-band satellite system is of increasing interest around the world due to its huge bandwidth. Rain fading is one of the primary factors affecting performance and availability of the Ka-band system. Extra power on the satellite can provide compensation for rain attenuation. In this thesis, we study the rain fade compensation problem for downlink transmission in the Ka-band satellite by dynamic resource allocation. The resources we consider include power and antennas onboard the satellite. The goal is to maximize the aggregate priority of packets arriving at all downlink spots as well as maintain fairness among downlinks. We formulate the problem mathematically in the framework of Knapsack Problems (KP). In particular, we show the resource allocation problem is equivalent to a Multi-choice Multiple Knapsack Problem (MCMKP), which, in general, is very hard to solve in a reasonable time. By introducing the seeding theory into the antenna scheduling, we decompose the original MCMCP into a sequence of Multiple-choice Knapsack Problems (MCKP), which are easier to solve.

The effectiveness of our approach is demonstrated through simulations in OPNET. Comparison with the Multiple Knapsack Problem (MKP) approach proposed by Birmani is also provided.

RESOURCE ALLOCATION IN KA-BAND SATELLITE SYSTEMS

by

Youyu Feng

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
Of the requirements for the degree of
Master of Science
2001

Advisory Committee:

Professor John S. Baras, Chair
Professor Armand Makowski
Professor Subramanian Raghavan

DEDICATION

To my family

ACKNOWLEDGEMENTS

I am grateful to my advisor Dr. John S. Baras for his advice, support and encouragement. I would also like to thank Dr. Armand Makowski and Dr. S. Raghavan for agreeing to serve on my committee and to review this thesis.

This thesis was motivated by Vineet Birmani's work. I am also grateful for his valuable help in the start of this work. Special thanks are due to Mingyan Liu, Manish Karir, Arvind Mani, Majid Raissi-Dehkordi and numerous other colleagues who contributed to constant help and support.

The research reported in this thesis was supported through collaborative participation in Advanced Telecommunications/Information Distribution Research Program (ATIRP) consortium sponsored by the U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL 01-96-2-0002. This support is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

1.1 Introduction to Ka-band Satellite Systems

There has been considerably increasing interest in expanding the broadband integrated services to include satellite communication links. Compared to conventional terrestrial networks, satellite communications have the following attractive features:

- Ubiquitous access: Services are available to whole regions within satellite footprints, including locations where terrestrial wired networks are not possible or economically infeasible.
- Broadcast/multicast nature: Many multimedia applications benefit from this feature of satellite networks.
- High bandwidth: Satellite channels today can deliver gigabits per second.
- Flexible bandwidth-on-demand capability: This may result in maximum resource utilization.

To provide sufficient bandwidth to meet the growing demand for satellite transmission capacity, people need to exploit higher frequency range and develop new technologies. In the late 1970's, the Ka band (20/30GHz) was selected by many space agencies around the world as the frequency band for the next generation broadband satellite networks. Utilizing the Ka band and even higher frequency bands has obvious advantages over the lower frequency ones:

- Large bandwidth: Huge bandwidth available in this frequency range is the primary motivation for developing Ka band satellite systems.

- Small antenna size: The increasing radio frequency implies that we can decrease the size of the antenna beam shape. Thus, either the distortion due to interference from adjacent satellite systems is reduced, or antennas with smaller diameter can be used. Smaller antenna size makes broadband satellite services affordable to millions of personal and commercial end-users.
- Even larger system capacity: Using many small spot beams in the Ka band systems increases the satellite power density and permits large frequency reuse, which leads to a much larger effective bandwidth. Thousands of user terminals equipped with inexpensive antennas can be served at the same time without using expensive hubs.

As early as in 1970's, researchers started to explore the Ka-band region in the United States as well as in Europe and Japan. The first Ka band satellite services were introduced with the basic technologies for transparent transponders in Japan.

The first operational regenerative Ka-band system integrated with terrestrial networks, was implemented in the Italian Ka-band program, ITALSAT. Since ITALSAT-F1 was successfully launched in January 1991, satellite has been no longer a "cable in the sky" based on transparent transponders; instead it has become a network node. Main features of the system included: Italian coverage obtained by means of six very narrow spot beams; total capacity of 0.9 Gbit/s achieved with 147 Mbit/s time division multiple access (TDMA) in the uplink; interspot connectivity provided by a synchronous baseband space-switch matrix; TDM in the downlink. ITALSAT system also provided operational experience for reallocation of capacity in a fast and flexible way [1].

In the United States, the Advanced Communications Technology Satellite (ACTS) program was formulated at the National Aeronautics and Space Administration (NASA) in 1984 to continue NASA's role of developing advanced space communications technology. The ACTS satellite was launched in September 1993. It created a revolution in the satellite system architecture by introducing the following key digital technologies in Ka-band systems [2]:

- Fast hopping multibeam antenna
- On-board baseband processor
- Wide-band microwave switch matrix
- Adaptive rain fade compensation
- Very small and ultra small aperture terminal
- High data rate terminal and 900 MHz transponder

These technologies have become the foundation of the current interests in the use of Ka band in global interactive multimedia systems.

Stimulated by the strong industrial interest, the Federal Communications Commission (FCC) awarded 13 licenses for the use of Ka band in the United States in 1997. Hughes' SPACEWAY was among the first filed systems. The SPACEWAY network is aimed at providing interactive "bandwidth-on-demand", cost-effective, multimedia communication services for hundreds of millions of people within the continuous view of the satellites. The state-of-the-art features of SPACEWAY network are listed below [3]:

- Narrow (about 1°) and wide (3°) spot beams cover both populated and low population areas.

- On board processors and switches provide individual customers with immediate access to the satellite, route packets within appropriate spot beams, and interconnect with other satellites in the network.
- Small, easily installed ground terminals bring satellite technology to the economic threshold of a greater universe of customers.
- Various digital transmission bit rates can support a variety of applications.

Also, through a unique arrangement of intersatellite links, SPACEWAY, which was proposed to launch in the time window 2002-2003, will create the first truly interconnected worldwide network.

1.2 Motivation for Resource Management

Most new generation Ka-band satellite systems like SPACEWAY are being designed to provide low-cost telecommunication services to hundreds of millions of users. Thus efficient management of various satellite and spectrum resources is required to meet the fast-growing service demand. Some of these resources, like the frequency spectrum, have been a limited factor in most of the old and present day systems, so a lot of work has been done in designing good resource allocation algorithms. Allocation of satellite power and antennas gained less attention in the past. But it has become more and more important because of the special rain fade problem and new technologies such as multibeam antennas in satellite systems operating at Ka band.

1.2.1 Rain Fade Problem

Having the advantages of increased bandwidth and significantly smaller ground terminal equipment, Ka frequency band was long maligned as being totally impractical for use by satellite. The “bad” mask was the degradation due to atmospheric propagation effects which is much more severe than those found at lower frequency bands.

The primary propagation factors that affect Ka-band earth-satellite channels include:

- Rain attenuation
- Wet antenna
- Depolarization due to rain and ice
- Gaseous absorption
- Cloud attenuation
- Atmospheric noise
- Troposphere scintillation

Among all these factors, rain fade presents the most challenging impediment to system designers because signal attenuation due to rain is the most severe propagation effect at Ka band. According to ACTS’ propagation research, rain attenuation at 20 GHz is almost three times that at 11 GHz and it can easily exceed 20 dB in many areas of the world.

Rain attenuation is a function of frequency, rain intensity, raindrop size distribution, raindrop temperature, elevation angle and polarization angle. For example, the relationship between frequency and rain attenuation is approximately as follows:

$$\frac{A_1}{A_2} \approx (\frac{f_1}{f_2})^2,$$

where f_i, A_i $(i=1,2)$ represent the frequency and the corresponding attenuation, respectively.

The following rain fade characteristics need careful consideration in fade compensation [5]:

- Rain time: In general, the average rain time that needs compensation is less than 5%-10% of a year. Thus dynamic resource allocation would be better than fixed link margins.
- Simultaneous rain fade over extended areas: A preliminary analysis indicates that fades at sites separated by distances exceeding the average rain cell size are uncorrelated.
- Fade rates: Rain fade rates rarely exceed 1 dB/s for most locations.
- Fade duration: Fade duration varies from several seconds to a few hours depending on the system margin and rain conditions.
- Frequency scaling: Uplink and downlink fades are generally correlated. Thus accurate fade measurements in only one direction are enough for fade compensation.

The downlink rain attenuation can be measured directly by observing the power of the 20 GHz beacon signal received at the earth station. Frequency scaling techniques then can be used to compute the fading in the 30 GHz uplink. When rain fade is determined, appropriate methods can be implemented to mitigate the fade.

1.2.2 Review of Rain Compensation Approaches

Satellite communication systems operating at Ka-band are subject to impairments produced by the troposphere, especially the rain attenuation. As a consequence, fade compensation schemes have to be implemented to guarantee certain system performance and availability. During the past few years, considerable effort has been devoted to developing effective fade mitigation techniques. Roughly speaking, there are four different approaches.

The most intuitive approach would be using larger ground station antennas and/or higher power amplifiers. But since the current trend is to use small (< 20 inches apertures), low-cost ground terminals (< $1000) that are affordable by a great universe of customers, this form of compensation would be too expensive to most end-users. In addition, since the average rain time for which compensation must be employed is usually short (< 10%), the added system margin will be wasted for over 90% of the time.

Site diversity is another effective but "expensive" countermeasure in combating rain fade. This technique involves tandem operation of two earth stations located several kilometers apart in distance. As we mentioned before, rain fades at sites separated by distances exceeding the average rain cell size (several kilometers) are expected to be uncorrelated. This enables a re-routing of the traffic via the less affected earth station whenever a severe attenuation occurs at the other site. But the cost of two earth terminals makes this approach not applicable to common customers' budget.

The third approach is to provide additional power to the transmit carriers at the satellite to compensate for rain attenuation. As the downlink rain fading occurs in some beam, power control correction of approximately 1.5 times fade is required to maintain the carrier to noise ratio. Transponders with various output power levels that are necessary for this mitigation method should be commanded into high power modes under rain conditions and switch back whenever the fading is over.

The most well known approach is ACTS' adaptive rain fade compensation. This protocol provides 10 dB of margin by reducing the burst by half and invoking one-half-rate forward error correction coding during a period of signal loss caused by rain. This protocol also includes a decision process, which makes use of the downlink signal level together with the FADED and CLEAR thresholds identified for each very small aperture terminal (VSAT) to determine the need for compensation in real time.

The third and fourth compensation techniques both try to alleviate rain impairments by setting aside an extra portion of system capacity. These resources will be allocated to beams suffering from rain attenuation only when needed. For example, in Time Division Multiplex (TDM) systems, the additional time slots will provide adequate redundancy for impaired signals.

Based on the third approach, Birmani proposed a power allocation and antenna scheduling scheme in his thesis [4]. The basic idea there was to boost the power of beams under rain conditions to maintain the normal bit rate, and then schedule bursts in such a way that the aggregate profit is maximized. In particular, he posed the scheduling problem as a multi-knapsack problem (MKP).

1.3 Contributions and Organization

Motivated by Birmani's work, this thesis proposes an effective and flexible rain fade compensation scheme. We first model the rain fade compensation as a linear integer programming problem, and further formulate it in a framework of “multi-choice multiple knapsack problem” (MCMKP). This framework subsumes Birmani’s MKP model as a special sub-solution. Completely solving the MCMKP in reasonable time is intractable in consideration of the number of variables involved. Then we present a sub-optimal scheme to the original optimization problem, which decomposes the MCMKP into a sequence of multi-choice (single) knapsack problems (MCKP). The latter is solvable in real time. To be specific, our scheme consists of two parts: scheduling antennas using the seeding theory, and allocating power by solving MCKP. Essentially our approach decouples the originally coupled antenna scheduling problem and power allocation problem. Compared to the MKP scheme, our MCKP scheme enjoys the following advantages: fairness, maximum utilization of extra power, and low computation complexity. The effectiveness of our resource allocation scheme is demonstrated by simulation in OPNET.

The remainder of the thesis is organized as follows. Chapter 2 describes the system configuration, states the problem we want to solve and gives the mathematical formulation. In Chapter 3, the classical theory of knapsack problems (KP) is briefly reviewed and several variants of KP relevant to our problem are introduced. We investigate the relationship between the rain fade compensation and knapsack problems followed by the detailed description of our multi-choice knapsack allocation scheme in Chapter 4. Chapter 5 provides the simulation implementation and results. Finally, conclusions and suggestions for future work are given in Chapter 6.

Chapter 2: Problem Description and Formulation

2.1 System Configuration

In this work, we will focus on the geosynchronous Earth orbit (GEO) satellites operating at Ka band and providing broadband services. Figure 2.1 below illustrates the typical satellite network architecture.

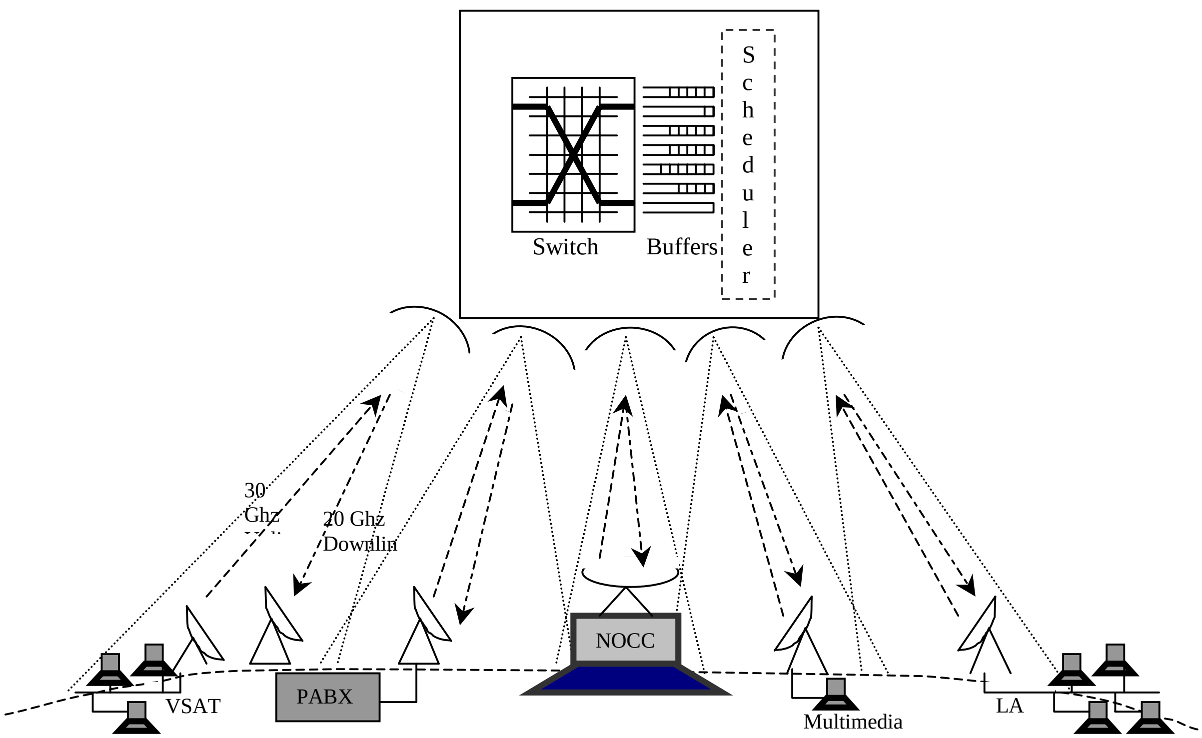


Figure 2.1: Typical satellite network architecture

The considered network scenario is mesh configured, comprising a satellite with on-board switching/processing, hundreds of low-cost earth stations generating different types of traffic and a Network Operations and Control Center (NOCC) that collects data, exchanges information among the network components and controls the operations of the satellite.

2.1.1 Multimedia Services

Using Ka band satellite, interactive multimedia services can be provided globally to fixed and mobile users with inexpensive cost. Various applications supported by the system include: internet web browsing, bulk date transfer, interactive on-demand and database consultation, voice, video conference, image transmission, etc.

In the context of integrated services networks, we consider four distinct service categories [6], [7]:

- *Guaranteed Service (GS)*: This category includes the real time and long-lasting calls which require low packet loss and minimum delay. ATM classes CBR (constant bit rate) and rtVBR (real-time variable bit rate) can be mapped into this category. GS has the highest priority.
- *Sustainable Service (SS)*: This category requires only low packet loss. ATM class nrtVBR (non-real-time variable bit rate) falls into this category. SS has lower priority than GS.
- *Controlled Service (CS)*: ATM class ABR (available bit rate) belongs to this family. CS can tolerate slight packet loss and bounded delay. Its priority is lower than SS.
- *Best Effort (BE)*: This category corresponds to ATM UBR (unspecified bit rate) service class. It requires no guarantee and has the lowest priority.

According to the different QoS (Quality of Service) requirements of the above categories, we assign priorities 4, 3, 2, 1, to the traffic belonging to GS, SS, CS and BE respectively.

2.1.2 Uplink and Downlink

The number of downlink spots in the system is about four times that of uplink spots. Consequently, the downlink cell size is much smaller than that of the uplink. Thus, downlink power is concentrated and small antennas are allowed.

The earth stations share the 30GHz uplink (earth to satellite) channel in a Multiple Frequency TDMA manner (MF-TDMA) [7], which combines Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA). The total bandwidth allocated to each spot beam is first divided into a number of non-overlapping carriers, as the rows in Figure 2.2. This allows for the smaller size of the ground stations due to the lower transmission rates. Then each sub-channel is further divided into non-overlapping time slots, as the columns in Figure 2.2. This combination of FDMA and TDMA makes the bandwidth utilization more flexible and efficient.

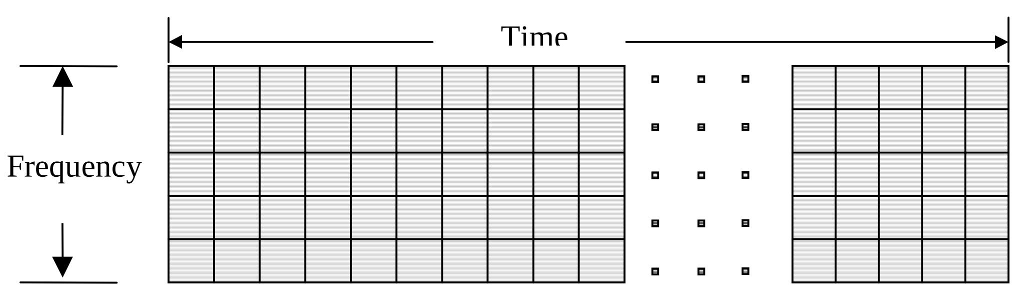


Figure 2.2: Uplink MF-TDMA scheme

On the 20 GHz downlink (satellite to earth), the access mechanism inside every spot is time division multiplex (TDM).

In this thesis, we will focus on the resource allocation problem in the downlink transmission. There are tens of antennas and hundreds of downlink buffers on the satellite. Downlink transmission to the ground spots is organized into bursts, each of which occupies a fixed time interval. Each antenna serves one and only one downlink spot during a burst. To guarantee certain Bit Error Rate (BER) performance, the maximum downlink transmission rate *B* allowed is a function of the transmission power and the weather condition:

$$B = f(power, rain),$$

as illustrated in Figure 2.3. To be specific, for a fixed transmission power level, we need to reduce the transmission rate to satisfy the BER requirement when rain condition gets worse. On the other hand, under the same weather condition, with a higher power level, we can raise the transmission rate without affecting the BER performance.

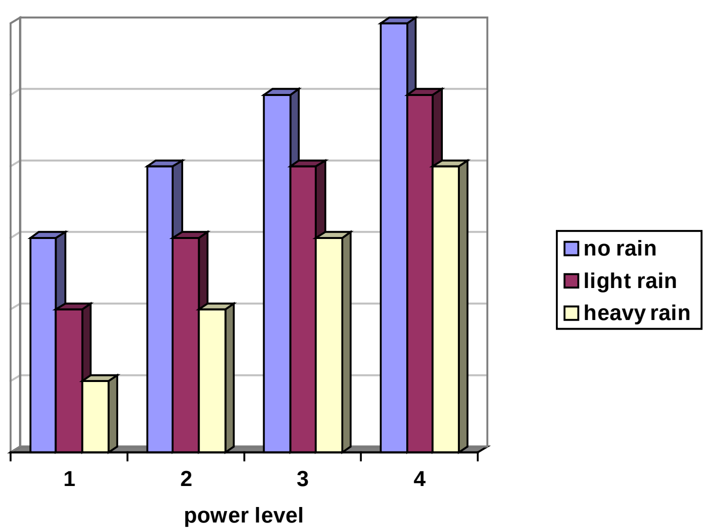


Figure 2.3: The transmission rate vs. power level and rain condition

For convenience of discussion, we will fix certain BER requirement in the sequel. Also we define certain transmission rate as the *standard rate*. The corresponding *standard power* for each downlink is thus defined to be the power required serving this downlink at the standard rate under clear weather condition. We assume the satellite has some extra power in addition to the sum of standard power needed by downlink spots in a burst, which provides compensation when some downlinks suffer from rain fade. In particular, we will assume that with the extra power, the standard rate can still be maintained if the rain area is less than 10%.

The antennas can adjust power levels and thus transmission rates to accommodate weather conditions. The earth stations are also capable of doing appropriate adjustment.

2.1.3 Onboard Switch and Scheduler

Due to the large number of beams, an onboard switch is required to route traffic among the end spot beams. Since the number of uplink beams is different from the number of downlinks, the switch matrix would be asymmetric, that means, the switch has unequal number of input and output ports.

The onboard scheduler will receive control information from the Network Operations and Control Center (NOCC), pick the appropriate downlink beams, allocate power to these beams and schedule the bursts.

2.1.4 Network Operations and Control Center (NOCC)

NOCC is the core of this network scenario. It instructs the satellite to operate in different modes according to the information it collects. The resource allocation work will be done mainly in NOCC.

The typical mission lifetime for a Ka band satellite will be 10-15 years. During this period of time, the Internet traffic will grow even faster and the types of applications will change unpredictably. A preprogrammed algorithm onboard the satellite will not be able to provide efficient capacity allocation and utilization, thus implementing the resource management algorithms (which maybe change as the time evolves) in NOCC on the ground would be a better choice.

For rain fade compensation, each earth terminal measures the downlink signal level and transmit it to the NOCC. Taking the reported signal level as an input, we determine the corresponding rain fade condition through a hysteretic operator [8], as illustrated in Figure 2.4. The rain fade condition takes value from {Heavy rain, Light rain, No rain}. If the signal level is between the predefined CLEAR1 and FADE2 thresholds, the earth terminal is claimed to be under light rain condition. The rain condition remains “Light rain” until the signal increases and passes CLEAR2, or it decreases and passes FADE1. In the first case, we say the rain fade is over; while in the second case, we claim the terminal is suffering from heavy rain fade. Similarly, we can determine the fade levels for various other cases.

Using both CLEAR and FADE thresholds we can cope with the noise in signal level measurement and add stability to the decision system. These thresholds are set individually for every earth terminal based on their BER performance.

NOCC also collects the traffic data, such as traffic type and traffic load, from the ground stations and the satellite. Whenever substantial change in traffic occurs or an earth terminal requires rain fade compensation, NOCC will call the resource allocation algorithm and transmit the resulting operational schedule to the satellite. The scheduler onboard the satellite thus manages its next burst according to the new schedule.

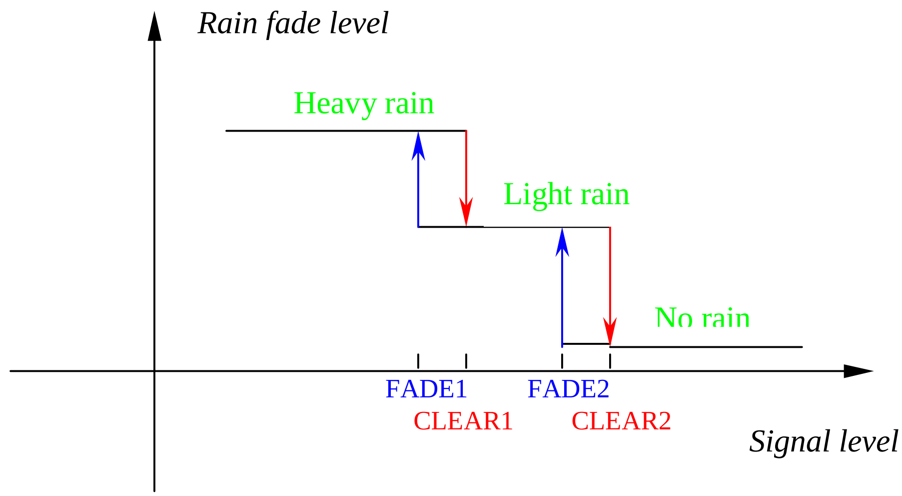


Figure 2.4: Hysteretic relationship between signal level and rain fade condition

2.2 Problem Description

In this section, we will state the resource allocation problem in the scenario described in the previous section.

We have defined priorities for different types of services in Subsection 2.1.1. Thus every uplink packet has a priority set in its header. Every downlink spot beam is assigned an individual buffer on the satellite. After receiving the packets, the onboard switch will route them into appropriate downlink buffers according to the destination addresses specified in their headers.

As we mentioned earlier, the number of downlinks is many (say 30) times that of antennas, and each antenna serves one downlink spot during a burst. The resources we consider here include antennas and the total power of antennas. By resource allocation, we mean two things:

- *Burst scheduling*: Assignment of antennas to downlinks for each burst period;
- *Power allocation*: Allocation of power to each antenna under the constraint that the total power of antennas does not exceed a specified limit.

Our objective in the resource allocation is two-folded: high profit and fairness, which are made clear below.

- *High profit*: We define the profit by the aggregate priority collected at all earth stations during a fixed time interval (to be specified soon), i.e., the sum of priorities of all packets received at all terminals.
- *Fairness*: We want to prevent the following situation from happening: one or more downlinks do not get service for a relatively long time.

In consideration of this fairness requirement, we define the time interval during which high profit is sought, to be the time it takes to serve every downlink one and only one burst with no antenna idling. In the sequel, we call this time interval "a round", and it is the time horizon for our resource allocation problem.

If there were no rain fade or traffic variation, the solution is straightforward: serving the downlink buffers in a round robin manner with a fixed data rate. This scheme is very simple and fair to every downlink. Unfortunately, this is not the real case.

When rain fades occur in some spot beams, those spots may not be able to be served with the fixed rate due to the limited total transmission power in satellite. In

Subsection 2.1.2, we have described the relationship between the transmission rate, the transmission power, and the rain fade level for a certain BER performance. Under certain rain condition and BER requirement, if there is extra power available, we can raise the transmission power to hold the fixed rate, otherwise we have to reduce the transmission rate. In other words, when the rain condition and required BER performance are given, there is only one freedom left, either power or rate, for each downlink. Since the transmission power is the active factor in these two, we view it as a power allocation problem.

From the above analysis, we can see that the antenna assignment and power allocation problems are coupled in that antenna assignment cannot be done without considering the power settings for the selected downlink buffers and vice versa. Thus these two problems must be considered together to achieve high profit.

In short, the resource allocation problem can be stated as follows:

For each burst period in one round, we want to select downlinks to be served and allocate associated transmission power to them within the constraint of total available power, so that under various weather condition distributions, the aggregate profit is maximized and the fairness requirement is satisfied.

2.3 Problem Formulation

In this section, we will give the mathematical formulation of the resource allocation problem.

2.3.1 Notation

First, we introduce the notations that will be used in the remainder of the thesis.

N — number of antennas;

M — number of downlink spots (buffers);

L — number of bursts in a round,

$L = M / N$;

R — number of transmission power levels for every downlink spot;

P_{tot} — total available power for each burst;

w_{mr} — transmission power of level *r* for downlink *m*, with higher *r* indicating higher power level.

$m = 1, 2, \cdots, M$, $r = 1, 2, \cdots, R$;

d_{mr} — number of packets that can be transmitted in downlink *m* in one burst time using transmission power w_{mr} under current rain condition.

$m = 1, 2, \cdots, M$, $r = 1, 2, \cdots, R$;

p_{mr} — priority sum of the first d_{mr} packets in buffer *m*,

$m = 1, 2, \cdots, M$, $r = 1, 2, \cdots, R$;

x_{lmr} — indicator of whether the *m*th downlink spot with power level *r* is allocated to the *l*th burst,

$l = 1, 2, \cdots, L$, $m = 1, 2, \cdots, M$, $r = 1, 2, \cdots, R$,

$$x_{lmr} = \begin{cases} 1 & if\ spot\ m\ with\ power\ level\ r\ is\ put\ in\ burst\ l \\ 0 & O.W. \end{cases}$$

The following constraints regarding the above parameters and variables are satisfied in practice:

(1) $M >> N$ and M is integer divisible by N.

(2) Consider the power matrix as follows:

$$P = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1R} \\ w_{21} & w_{22} & \cdots & w_{2R} \\ \vdots & \vdots & & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{MR} \end{bmatrix}$$

The differences of w_{mr}, $m = 1, 2, \cdots, M$ inside each column are much smaller than the differences of w_{mr}, $r = 1, 2, \cdots, R$ inside each row.

(3) $\sum_{m=1}^{M} w_{m1} < LP_{tot} < \sum_{m=1}^{M} w_{mR}$. As we mentioned in Subsection 2.1.2, the total system power is enough to provide standard power (which is higher than the minimum power w_{m1}) to every downlink in a burst, while it cannot supply everybody with the highest power w_{mR} (which is higher than the standard power).

2.3.2 Mathematical Formulation

The coupled resource management problem of antenna scheduling and power allocation is formulated as follows: given the rain fade condition of every downlink spot and the BER requirement,

$$\text{maximize} \qquad \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{r=1}^{R} x_{lmr} \, p_{mr} \,,$$

$$\text{subject to} \qquad \sum_{m=1}^{M} \sum_{r=1}^{R} x_{lmr} w_{mr} \le P_{tot} \,, \qquad l = 1, 2, \cdots, L \,, \qquad (1)$$

$$\sum_{m=1}^{M}\sum_{r=1}^{R} x_{lmr} \le N , \qquad l = 1, 2, \cdots, L , \qquad (2)$$

$$\sum_{l=1}^{L}\sum_{r=1}^{R} x_{lmr} = 1 , \qquad m = 1, 2, \cdots, M , \qquad (3)$$

$$x_{lmr} \in \{0,1\}, \qquad l = 1, 2, \cdots, L ,\ m = 1, 2, \cdots, M ,$$

$$r = 1, 2, \cdots, R .$$

The first constraint ensures that the system power is enough to serve the selected spots with their respective power in each burst. The limit of antenna number is represented in constraint (2). Constraint (3) guarantees fairness among downlink spots by serving every spot once and only once in a round.

All the numbers p_{mr}, w_{mr}, N and P_{tot} are positive integers. And also the objective functional and the constraints are linear in x_{lmr}, thus the above problem falls into the class of (linear) integer programming. In particular, it has the similar structure as the well-known 0-1 knapsack problems, which we will discuss in detail in the next two chapters.

Chapter 3: Knapsack Problems: Some Background

3.1 Introduction to Integer Programming

A linear program is a mathematical model designed to find a set of decision variables to maximize (or minimize) a linear objective function while satisfying some linear constraints. If the restriction that decision variables must take integer values is added, we have a (linear) *integer program* (IP) [9], [10], [11].

An integer programming problem can be formulated as:

$$\begin{aligned} &\text{maximize} && cx, \\ &\text{subject to} && Ax \le b, \\ &&& x \ge 0 \quad , \\ &\text{and} && x \text{ integer}, \end{aligned}$$

where A is an m by n matrix, c an n-dimensional row vector, b an m-dimensional column vector, and x an n-dimensional column vector of decision variables. And if all variables are further restricted to 0-1 values, we have a *0-1 or binary integer program* (BIP):

$$\begin{aligned} &\text{maximize} && cx, \\ &\text{subject to} && Ax \le b, \\ &&& x \in \{0, 1\}^n . \end{aligned}$$

A wide variety of practical problems can be formulated as or converted to integer programs. Included in these are scheduling, planning, location, network, cutting and selection problems that arise in industry, military, education, health, and other environments. In the past ten years, there has been a remarkable advance in the integer programming field due to improved modeling, faster computers, new cutting plane theory, branch-and-cut and other advanced algorithms. So more complex problems can be modeled and solved using integer programming in a reasonable computing time [9].

3.2 Overview of 0-1Knapsack Problems

An important class of binary integer programming problems is the family of 0-1 *knapsack problems* (KP). The name is in reference to packing a knapsack (or knapsacks) by choosing a subset of the given n items such that the corresponding profit sum is maximized without exceeding the capacity of the knapsack(s). The decision variable x_j is either 1 (item j is selected) or 0 (item j is not selected).

Knapsack problems have been extensively studied during the last three decades with a rich literature (see Pisinger [14], Martello and Toth [12] and Lin [13] for great surveys). The KP family is one of the widely discussed topics in integer programming mainly because of the following two reasons:

- Their immediate applications in industry and financial management such as budget control, project selection, cargo loading, and cutting stock.
- They appear as sub-problems in various integer programming algorithms. Many complex combinatorial optimization problems can be reduced to knapsack problems and they benefit from improvements in the field of knapsack problems.

Different types of 0-1 knapsack problems occur while various distributions of the knapsacks and items arise: In the *0-1 Single Knapsack Problem* (SKP) only one knapsack needs to be filled and each item may be chosen at most once; Special case of *Subset-sum Problem* arises when for each *j*, the profit c_j equals the weight a_j; If the items should be chosen from disjoint classes and exactly one item from each class, we obtain the *Multiple-choice Knapsack Problem* (MCKP); The *Multiple Knapsack Problem* (MKP) occurs when several knapsack of (maybe) different capacities are to be packed simultaneously.

The generalizations of 0-1 knapsack problems include the *Bounded Knapsack Problem, Unbounded Knapsack Problem and Bin-packing Problem.* If the amount of items chosen from each item type is unlimited or bounded by a finite number, we get the *Unbounded or bounded Knapsack Problem* respectively. The *Bin-packing problem*, which is designed to pack all items into minimum number of equally sized bins, is an example of minimization problem.

The most general form of a knapsack problem is the *Multidimensional Knapsack Problem*, also known as *Multi-constrained Knapsack Problem.* While it has the formulation of general integer programming, all the coefficients in the object function and constraints are required to be nonnegative.

All Knapsack problems belong to the *NP-hard* family (see Garey and Johnson [15]), therefore it is very unlikely that polynomial time algorithms can be devised for them. The only way to get an exact solution is an enumeration in the solution space. If the enumeration is complete, unacceptable solving time is expected. Fortunately, several effective enumerative techniques have been developed during the past decades of research to save quite a lot of efforts [9], [16], [17]:

- *Branch and bound*: build an enumeration tree, and remove the nodes which cannot produce improved solutions by using bounds derived from the integrality, nonnegativity, and other constraints. This is also called *implicit enumeration.*
- *Preprocessing*: before solving the program, quickly check the “sensibility” of the formulation, detect and eliminate redundant constraints and variables, and tighten bounds where possible.
- *Dynamic programming*: calculate the optimal solution recursively from the optimal values of slightly different problems.
- *State space relaxation*: Scale the coefficients by a fixed value. In this way the time and space complexity of an algorithm may be considerably decreased, at the loss of optimality. Several efficient algorithms arise from state space relaxation.

In the next several sections, we will give more detailed descriptions of some well-developed knapsack problems which are most related to our work.

3.3 0-1 Single Knapsack Problem

The most fundamental knapsack problem is the 0-1 single knapsack problem (SKP). Given *n* items, each with weight w_j and profit p_j, and a knapsack with capacity *c*, the problem is to fill the knapsack so that the profit sum of the chosen items is maximized and the weight sum of these items does not exceed the knapsack capacity. 0-1 single knapsack problem can be described mathematically as:

$$\text{maximize} \qquad z = \sum_{j=1}^{n} p_j x_j \,,$$

$$\text{subject to} \qquad \sum_{j=1}^{n} w_j x_j \le c \,,$$

$$\text{where} \qquad x_j = \begin{cases} 1 & \textit{if item } j \textit{ is selected} \\ 0 & O.W. \end{cases} \qquad j = 1, 2, \cdots, n \,.$$

Without loss of generality, we make the following assumptions about the coefficients p_j, w_j, and c:

(1) All coefficients are nonnegative integers; fractional case can be transformed by multiplying some factor.

(2) $p_j > 0$: Otherwise it can be removed from the item set.

(3) $0 < w_j < c$: 0-weight item can be directly put into the optimal solutions and items with weight exceeding c can be deleted.

(4) $0 < c < \sum_{j=1}^{n} w_j$: We can get trivial solutions by setting all $x_j = 0$ for case $c = 0$ and all $x_j = 1$ for case $c \ge \sum_{j=1}^{n} w_j$.

SKP is representative of many industrial situations such as budget control, cutting stock and project selection. It also appears as a sub-problem in many algorithms of other integer programming and knapsack problems: the multiple knapsack problem, to mention an example.

SKP is *NP-hard*, but it can still be solved in pseudo-polynomial time. The problem has been intensively studied since 1966 due to its wide applicability and theoretical interest. See Dudzinski and Walukiewicz [18] (the theoretical framework of exact algorithms), Martello and Toth [12] (elaboration and implementations of these algorithms) and Gerasch and Wang [19] (parallel computing methods) for thorough reviews.

3.4 Multiple Knapsack Problem

The multiple knapsack problem (MKP) deals with packing *m* distinct knapsacks with *n* given items. The *m* knapsacks have (maybe) different capacities c_i, $i = 1, 2, \cdots, m$. Each item has a profit p_j and the associated weight w_j, and the problem is to choose *m* disjoint subsets from the *n* items, such that the total profit sum of the selected items is maximized while the weight sum of subset *i* does not exceed the capacity of knapsack *i*, for each $i \in \{1, 2, \cdots, m\}$. The multiple knapsack problem thus can be formulated as:

maximize $$z = \sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} ,$$

subject to $$\sum_{j=1}^{n} w_j x_{ij} \le c_i , \qquad i = 1, 2, \cdots, m ,$$

$$\sum_{i=1}^{m} x_{ij} \le 1 , \qquad j = 1, 2, \cdots, n ,$$

where $$x_{ij} = \begin{cases} 1 & \textit{if item } j \textit{ is assigned to knapsack } i \\ 0 & O.W. \end{cases}$$

$$i = 1, 2, \cdots, m , \quad j = 1, 2, \cdots, n .$$

Without loss of generality, we will make similar assumptions as in single knapsack problems to avoid trivial cases:

(1) All the coefficients p_j, w_j, and c_i are positive integers.

(2) $\sum_{j=1}^{n} w_j > \max\{c_i \mid i = 1, 2, \cdots, m\}$. This avoids the trivial solution of putting all items in one knapsack.

(3) $w_j \le \max\{c_i \mid i = 1, 2, \cdots, m\}$ for $j = 1, 2, \cdots, n$. This ensures that every item can fit into at least one knapsack as otherwise it can be removed from the item set.

(4) $c_i \ge \min\{w_j \mid j = 1, 2, \cdots, n\}$ for $i = 1, 2, \cdots, m$. The knapsack violating this assumption can be taken out as it cannot contain any item.

MKP has an immediate application in cargo loading problems, e.g., loading *m* vessels/container with an optimal plan such that maximum benefit is achieved.

MKP is *NP-hard* in the strong sense, thus dynamic programming approaches cannot be applied to MKP. As a result, most reported algorithms in the literature focused on branch and bound techniques: Hung and Fish [20], Martello and Toth [12], and Pisinger [14] to mention a few examples. Among these, the algorithm presented in Pisinger [14] is more efficient for large problem instances and is selected to solve MKP in our work (see Chapter 5).

3.5 Multiple-choice Knapsack Problem

The last well-known knapsack problem we will describe here is the multiple-choice knapsack problem (MCKP). We consider the problem of packing items from *k* disjoint sets $N_1, N_2, \cdots, N_k$ into some knapsack of capacity *c*. Each item *j* in class N_i has profit p_{ij} and weight w_{ij}. We want to select exactly one item from each set to pack in the knapsack such that the total profit sum of the chosen items is maximized, and the weight sum does not exceed the knapsack capacity. The multiple-choice knapsack problem thus may be formulated as:

$$\text{maximize} \qquad z = \sum_{i=1}^{k} \sum_{j \in N_i} p_{ij} x_{ij} ,$$

$$\text{subject to} \qquad \sum_{i=1}^{k} \sum_{j \in N_i} w_{ij} x_{ij} \le c ,$$

$$\sum_{j \in N_i} x_{ij} = 1, \qquad i = 1, 2, \cdots, k ,$$

$$\text{where} \qquad x_{ij} = \begin{cases} 1 & \text{if item } j \text{ in class } i \text{ is selected} \\ 0 & O.W. \end{cases}$$

$$i = 1, 2, \cdots, k , \quad j \in N_i .$$

Similarly, we make the following assumptions:

(1) All coefficients p_{ij}, w_{ij}, and *c* are positive integers.

(2) The *k* classes are mutually disjoint with size n_i, $i = 1, 2, \cdots, k$.

(3) $\sum_{i=1}^{k} \min\{w_{ij} \mid j \in N_i\} \le c < \sum_{i=1}^{k} \max\{w_{ij} \mid j \in N_i\}$. This avoids infeasible situations or trivial solutions.

MCKP has many applications: capital budgeting, menu planning, etc. An application in KP theory is transform of nonlinear KP to MCKP.

MCKP is also *NP-hard* since it contains SKP as a special case: each item in SKP can be viewed as a two-element class by adding a dummy item (*p*,*w*)=(0,0). However, due to its special structure, Dudzinski and Walukiewicz [18] showed that MCKP is solvable in pseudo-polynomial time. The problem has been intensively investigated during the last two decades and a number of algorithms were presented in literature. We mention several here as examples: Nauss [20], Sinha and Zoltners [21], Dyer, Riha and Walker [22], and Pisinger [14], among which we use the minimal algorithm in [14] to solve the MCKP in our work.

Chapter 4: Formulation of Resource Allocation as Knapsack Problems

4.1 Multi-choice Multiple Knapsack Model

In Chapter 2, we described the resource allocation problem we want to investigate and gave the mathematical formulation as follows:

$$\text{maximize} \qquad z = \sum_{l=1}^{L}\sum_{m=1}^{M}\sum_{r=1}^{R} x_{lmr}\, p_{mr} \ ,$$

$$\text{subject to} \qquad \sum_{m=1}^{M}\sum_{r=1}^{R} x_{lmr}\, w_{mr} \le P_{tot} \ , \qquad l = 1, 2, \cdots, L \,,$$

$$\sum_{m=1}^{M}\sum_{r=1}^{R} x_{lmr} \le N \ , \qquad l = 1, 2, \cdots, L \,,$$

$$\sum_{l=1}^{L}\sum_{r=1}^{R} x_{lmr} = 1 \,, \qquad m = 1, 2, \cdots, M \ ,$$

$$\text{where} \qquad x_{lmr} = \begin{cases} 1 & \textit{if spot m with power level r is assigned to burst l} \\ 0 & O.W. \end{cases} ,$$

$$l = 1, 2, \cdots, L \ , \ m = 1, 2, \cdots, M \ ,$$

$$r = 1, 2, \cdots, R \,,$$

L, *M*, *R*, and *N* are the numbers of bursts in a round, downlink spots, transmission power levels, and antennas, respectively, w_{mr} stands for the transmission power of level *r* for downlink spot *m* and p_{mr} denotes the corresponding priority, and P_{tot} is the total available power for each burst.

This problem is a binary integer program (BIP) as we defined in Section 3.1. We can further relate it to the family of knapsack problems by making the following observation:

L bursts can be viewed as *L* knapsacks with the same capacity P_{tot}, and every downlink spot with its different power levels and associated priorities as individual item class. To be specific, each item *r* in class N_m (downlink spot *m*) has a profit p_{mr} and weight w_{mr}. Thus the problem is to choose exactly one item from each class to pack in *L* knapsacks, such that the profit sum is maximized without exceeding any knapsack's capacity. From the above discussion, the resource allocation problem is equivalent to a non-standard knapsack problem, which we shall call *Multi-choice Multiple Knapsack Problem* (MCMKP). The "multi-choice" part is responsible for selecting an appropriate power lever for each spot, so this accounts for the power allocation aspect; while the "multiple knapsack" part corresponds to picking at most *N* spots for every burst in the round, so it accounts for the burst scheduling aspect. The resource allocation problem is thus reformulated as multi-choice multiple knapsack problem in the following form:

maximize $$z = \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{r \in N_m} p_{mr} x_{lmr} ,$$

subject to $$\sum_{m=1}^{M} \sum_{r \in N_m} w_{mr} x_{lmr} \le P_{tot} , \quad l = 1, 2, \cdots, L ,$$

$$\sum_{l=1}^{L} \sum_{r \in N_m} x_{lmr} = 1, \quad m = 1, 2, \cdots, M ,$$

$$\sum_{m=1}^{M} \sum_{r \in N_m} x_{lmr} \le N , \quad l = 1, 2, \cdots, L ,$$

where $$x_{lmr} = \begin{cases} 0 & if\ item\ r\ in\ class\ m\ is\ assigned\ to\ knapsack\ l \\ 1 & O.W. \end{cases},$$

$$l = 1, 2, \cdots, L\ ,\ m = 1, 2, \cdots, M\ ,\ r \in N_m .$$

All the assumptions for knapsack problems listed in Chapter 3 are naturally satisfied by this problem's engineering background described in Subsection 2.3.1.

MCMKP subsumes MKP and MCKP as two special cases: setting $L = 1$ and $M = N$ results in MCKP; while letting $R = 1$ reduces MCMKP to MKP (with slight modification). As we mentioned in Chapter 3, MKP is *NP-hard* in the strong sense, so is MCMKP. Therefore it rules out the existence of pseudo-polynomial algorithms or fully polynomial approximation schemes. There has been little effort devoted to the particular structure like MCMKP in the literature. The best algorithms published up-to-date take about a fraction of second to solve relatively large MKP instances and pseudo-polynomial time for MCKP. MCMKP is a combination of these two problems, thus it is very unlikely that an exact algorithm with a reasonable computing time (like seconds) can be devised based on today's techniques of integer programming. Since our work is more "engineering" rather than "theoretical", we are more interested in finding a feasible sub-optimal solution than a time-consuming optimal solution. Therefore, we will adopt some appropriate reductions and simplifications to the original problem and make it easier to solve.

In the next two sections, we will provide two schemes for solving the problem, which may be viewed as modifications of the above MCMKP model. In Section 4.2, we introduce and discuss Birmani's work [4], where he modeled the problem as MKP. In Section 4.3, we first state the performance measures for resource allocation schemes in

Ka-band satellite systems, then we present our new approach of MCKP, which is the main contribution of the thesis.

4.2 Multiple Knapsack Model

In his master thesis, Birmani proposed a set of schemes to solve the resource allocation problem for Ka-band satellite systems. The system configuration in his thesis is very similar as that in this thesis except that the transmission rate for downlink spots is fixed to some standard rate rather than tunable as in our setting. As a consequence, there is only one power level associated with each downlink under certain rain condition. Compared to the MCMKP model formulated in last section, this simpler configuration "removes" the items with non-standard transmission rates from each item set, eliminates the multiple-choice part of MCMKP, and reduces the problem to MKP structure which is more tractable. We will elaborate this below.

Birmani investigated allocation schemes for the two limited resources, power and antenna, separately. In consideration of different weather and traffic conditions, he discussed two cases, namely *stable load condition* and *unbalanced load condition* respectively, and proposed different burst scheduling and power allocation algorithms for each load condition.

4.2.1 Stable Load Condition

By Birmani's definition, stable load condition means that the transmission system on the satellite can serve all the traffic arriving at the satellite without overfilling the buffers and the whole system is stable.

By this stability assumption, the total system power is enough to meet the demand. So the power allocation scheme is very simple: allotting the appropriate power to the downlinks so that they can maintain the standard transmission rate under their rain conditions.

For burst scheduling, Birmani utilized one of the simplest generalized processor sharing schemes, which is a variant of *Weighted Round Robin* (WRR). The weight associated with a downlink queue is defined to be the priority sum of packets in that queue up to a *search depth,* which is equal to the number of packets that can be sent out in one burst by an antenna with the standard transmission rate. As soon as the weight of every queue is determined, all the queues can be ranked in the decreasing order of weights and the antennas will serve these queues in a round robin manner.

WRR and the “power-on-demand” described above formed Birmani’s resource allocation scheme under stable condition. This scheme has two obvious advantages: (1) simple to implement; (2) effective in allocating power and antennas to downlinks under stable load condition. But the disadvantage is also obvious: the system capacity is not fully used. As we mentioned before, the satellite systems are usually designed to carry extra power in addition to that required by downlinks under the clear weather condition. So under stable condition, the extra backup power is wasted.

4.2.2 Unbalanced Load Condition

Unbalanced load condition, on the other hand, refers to the situations whenever stable load condition is not satisfied. In this case, the system cannot provide sufficient capacity to transmit all the arriving traffic and overflows in buffers occur.

Under this load condition, the total system power is not enough to assign every downlink what they want. Either the requirements of some spots for more power are denied, or some other spots are sacrificed to gain more aggregate profit depending on the scheduling criterion.

The objective of Birmani's burst scheduling is to maximize the weight sum of the selected spots in a round time without exceeding the available system power for each burst. Here the weight of a downlink is defined as in the previous subsection. Each spot has a weight p_j and associated power requirement w_j. The total power available is P_{tot}. The problem is thus formulated as:

$$\text{maximize} \qquad z = \sum_{i=1}^{L} \sum_{j=1}^{M} p_j x_{ij} \,,$$

$$\text{subject to} \qquad \sum_{j=1}^{M} w_j x_{ij} \le P_{tot} \,, \qquad i = 1, 2, \cdots, L \,,$$

$$\sum_{i=1}^{L} x_{ij} \le 1 \,, \qquad j = 1, 2, \cdots, M \,,$$

$$\text{where} \qquad x_{ij} = \begin{cases} 1 & if\ item\ j\ is\ assigned\ to\ knapsack\ i \\ 0 & O.W. \end{cases} ,$$

$$i = 1, 2, \cdots, L \,, \quad j = 1, 2, \cdots, M \,.$$

Compared with the definition of multiple knapsack problem in Section 3.4, it is straightforward to realize that this burst scheduling formulation is exactly the same as MKP and thus much simpler than the MCMKP model in Section 4.1.

This simplification benefits from the simpler transmission configuration: either serve the spots with standard transmission rate or provide no service at all. The multi-choice part of MCMKP thus degenerates to simplest 0-1 choices. In other words, the power allocation aspect is separate from the burst scheduling aspect and the coupled resource allocation problem is decomposed into two sub-problems, making the problem easier to solve.

As soon as downlink spots are selected for each burst, the power allocation is straightforward: allocate power only to these “lucky” downlinks so that they are served with the standard transmission rate.

The MKP scheduling and “power-after-selection” constitute Birmani’s resource allocation scheme under unbalanced load condition. Moreover, MKP algorithm dominates the effectiveness of the whole scheme.

Birmani tried two algorithms for MKP in his thesis: Martello and Toth [12] exact algorithm and Martello and Toth [23] approximation algorithm. The exact algorithm is most suitable for small input sizes and uncorrelated items, which are not satisfied in our problem and he concluded that the exact algorithm would be unable to generate results within the desired time for this application. Then the approximation algorithm was implemented to give a close to optimal solution in a reasonable time.

Birmani proposed the MKP model for scheduling the antenna bursts and thus achieved a more efficient utilization of limited power when the system could not guarantee sufficient service to every downlink. More profit is achieved at the price of some downlinks getting no service, i.e. some power is taken from lower weight spots to higher weight spots during the allocation procedure. This is not quite consistent with the criterion of fairness within downlink spots. Also, although the computing time of the approximation algorithm is reasonable, it is still not fast enough to provide near real time response to the condition changes. In the next section, we will propose our resource allocation scheme, which emphasizes the multi-choice aspect of the MCMKP model.

4.3 Multiple-choice Knapsack model

As we analyzed before, the MCMKP model combines the two resource allocation problems together and is unlikely to be solved in a reasonable time. Thus some reduction must be made to the MCMKP model to simplify the problem. The most straightforward reduction is to remove either the multi-choice or multiple knapsack part in some way, thus decompose the original coupled problem into two separate problems: the power allocation problem and the burst scheduling problem. In Birmani's approach, the multi-choice part was removed and thus a complete MKP formulation arose. In the following, we will discuss the other possible approach: removing the multiple knapsack part from the MCMKP model. The resulting formulation has a complete MCKP structure corresponding to power allocation problem, while a heuristic algorithm using the seeding theory is proposed for the burst scheduling part.

Before presenting the new scheme, we will first discuss the performance measures for various resource allocation strategies, i.e., what objectives we want to achieve through the resource management.

4.3.1 Performance Metrics

For a resource allocation problem like the one we discuss in this thesis, here are the three metrics we think the most important when judging various approaches:

- *Fairness*

- *Efficiency*
- *Computing time*

By fairness, in general, we mean that every customer sharing the resource(s) is served in a fair way according to some criterion. In our particular application, we say a scheduling plan is fair if every downlink spot is served once during each round.

Efficiency is another consideration. High efficiency is interpreted here as high profit gain. In particular fully utilization of available resource(s) will lead to high efficiency.

Fast computation speed is desirable in real time applications. Computing time depends on the complexity of the algorithms, the programming techniques and the speed of the computing device etc, among which we will pay most attention to the first one.

These three factors are actually closely related. Sometimes fairness is guaranteed at the cost of efficiency, while higher efficiency may benefit from more complex formulation or finer granularity, which means more computing time. There are always some tradeoffs for researchers to make.

4.3.2 A New Multiple-choice Knapsack Scheme

In this subsection, we will describe a new resource allocation scheme based on the analyses we have done so far. This work is motivated by Birmani's research and is aimed at providing a more fair, more efficient and faster solution to power and antenna sharing problem in Ka-band satellite systems.

Fairness is the first consideration. In the MKP scheme, under unbalanced load condition, some downlink spots may be out of service for one or even more rounds just because some higher priority downlinks are covered by rain and need rain fade compensation. This is unfair in terms of our definition of fairness. The MKP setting is unable to tackle this problem. However, it can be circumvented by using the variable transmission rate service in our satellite transmission configuration. To guarantee fairness, every spot will get a *base service* for each round under whatever condition. The base service itself will vary with different load conditions. More detailed description of base services in different cases will be given later on.

We also notice that in Birmani's scheme, the two resources did not get fully utilized in either of the two conditions. To be specific, under stable load condition, the extra power is wasted; under unbalanced load condition, some antennas stay idling. Therefore efforts toward efficiency improvement would be rewarding.

Before elaborating our MCKP scheme, we will give a brief introduction to the seeding theory widely applied in tournaments. We will adopt the idea behind the seeding theory in the burst scheduling.

Seeding Theory

In the sports world, *elimination tournament*, also called *knockout tournament*, is a widespread form of competition. In an elimination tournament, teams (or individual competitors) play head-to-head matches with the loser eliminated from further competition and the winner progressing to the next round of competition. How should the organizer match the teams and schedule all the rounds of competitions? Obviously, the organizer wants to make a "fair" seeding to promote the spectator interest and thereby earn lots of television money. The situations such as the two strongest teams are paired in the first round of competition should definitely be avoided. Schwenk showed two axioms of tournament seeding in [24]:

- *Axiom DC: Delayed Confrontation.* Two teams rated among the top 2^j shall never meet until the field has been reduced to 2^j or fewer teams.
- *Axiom SR: Sincerity Rewarded*. A higher-seeded team should never be penalized by being given a schedule more difficult than that of any lower seed.

A standard seeding schedule is given for a 16-team tournament in Figure 4.1. Stronger teams are given smaller numbers.

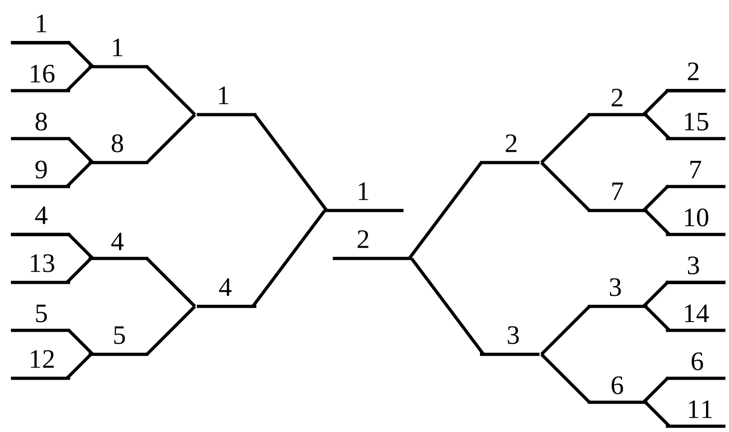


Figure 4.1: The standard method for seeding a tournament with 16 teams

In the round of size 2^r, numbers of each pair of opponents satisfy $i + j = 2^r + 1$. Actually, the pairings in the rounds except the first one will not necessarily occur. This figure just indicates the predicted pairings. This method clearly satisfies Axioms DC and SR.

Seeding theory can be used in many situations other than tournaments, for example, grouping students into classes, assigning people to different projects, etc. Similar idea is applicable to our burst scheduling problem.

Now we are ready to present the resource allocation scheme. In the following discussion, we will divide the entire scenario space into two cases: (1) the extra power is enough to compensate all the downlinks that are suffering from rain fades and thus the satellite system can provide at least standard service to everybody; (2) whenever case I is not satisfied, i.e., standard service cannot be guaranteed to every downlink. Case 1 and Case 2 here can be viewed as corresponding to the stable load condition and the unbalanced load condition respectively, in Birmani's thesis. But our classification is more straightforward and case identification is easier.

Our scheme will treat these two cases separately. In both cases, the primary steps of the assignment procedure are very similar, which are listed below:

- Step 1: Base service assignment
- Step 2: Burst scheduling through seeding
- Step 3: MCKP power allocation

The detailed implementation for each case, however, is different.

Case I: Rain fade area less than or equal to 10%

In this scenario, the total system power can provide standard transmission rate to every downlink. So the base service assigned to every downlink in Step 1 is based on the standard rate transmission, i.e., virtually assigning every spot whatever power they need to keep the standard transmission rate under current weather condition. Assignment in Step 1 is not really implemented, and it is only for conceptual illustration. The final power really assigned to every downlink, which is never less than that required for standard rate transmission, will be determined in Step 3.

In Step 2, downlink buffers are first sorted in a non-increasing order of their average priorities per packet. Afterward the ordered downlink spots are assigned to bursts in the following manner: the first spot goes to the first burst, the second one to the second burst, ⋯, the Lth buffer to the Lth burst, then the $L+1$st spot to the Lth burst, the $L+2$nd spot to the $L-1$st buffer, and so on. This scheduling plan is illustrated in Figure 4.2. The arrows indicate the order in which spots are filled into bursts.

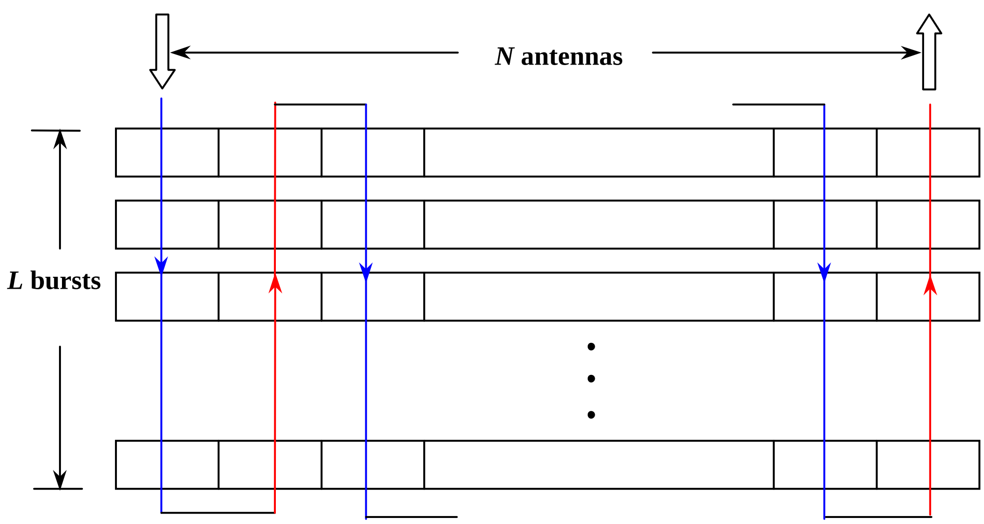


Figure 4.2: Illustration of burst scheduling

Seeding theory is applied here. Downlink buffers inside the same burst will compete for the total available transmission power. When packing downlinks into L

bursts, we certainly do not want to group all the highest weighted buffers together and grant additional power to few of them, while leave the additional power in other bursts to some lower weighted buffers. The underlying reason is very straightforward: same power can gain more profit when it is granted to buffers with higher priorities than to those with lower priorities. We shall name this burst scheduling scheme as *seed burst scheduling* in this thesis.

Intuitively, the seed burst scheduling scheme distributes the total profit that we want to maximize in MCMKP model into L bursts (nearly) evenly. Thus the original problem (MCMKP) is transformed to the problem of maximizing the profit of each burst, which, as we will discover shortly, is a multiple-choice knapsack problem. In short, the original complex MCMKP model is decomposed into L multiple-choice knapsack sub-problems, which are much easier to solve, thanks to the seed burst scheduling.

Step 3 is to allocate the power inside each burst. Since the problem is exactly the same for each burst, without loss of generality, we will focus on the first burst. Let's assume in Step 1, the nth downlink buffer in the first burst was assigned power w_{nr_n}, where $1 < r_n \le R$. Then each of the N buffers can be considered as an item set consisting of $R - r_n + 1$ items $(p_{nr_n}, w_{nr_n}), \cdots, (p_{nR}, w_{nR})$, where p_{nr}, $r_n \le r \le R$, is the priority of item r in the n-th item set S_n and w_{nr} is the associated power. Given a knapsack with capacity P_{tot} and N item sets $S_1, S_2, \cdots S_N$, the problem is to select exactly one item from each item set to pack in the knapsack. This is a standard multiple-choice knapsack problem and thus can be formulated as:

$$\text{maximize} \qquad z = \sum_{n=1}^{N} \sum_{r \in S_n} p_{nr} x_{nr} \ ,$$

subject to $\sum_{n=1}^{N} \sum_{r \in S_n} w_{nr} x_{nr} \le P_{tot}$,

$$\sum_{r \in S_n} x_{nr} = 1, \qquad n = 1, 2, \cdots, N,$$

where $x_{nr} = \begin{cases} 1 & if \quad item \quad r \quad in \quad set \quad S_n \quad is \quad selected \\ 0 & O.W. \end{cases}$,

$$n = 1, 2, \cdots, N, \; r \in S_n,$$

and $S_n = \{ r_n, \cdots, R \}$.

This formulation guarantees the base service to every item set (buffer) by providing at least w_{nr_n} to buffer *n*, while distributing the remaining power among higher priority buffers to gain more profit.

Case II: Rain fade area more than 10%

In this scenario, the total rain fade is so severe that it cannot be completely compensated by the extra power the system has prepared. As a consequence, we can not guarantee every downlink buffer to be served at the standard transmission rate. In this case, the base service provided to buffer *n* in Step 1 is thus based on the lowest power level w_{n1}, which is also called *base power* and is always lower than the standard power.

In Step 2, buffers are ordered and assigned to bursts using the seed burst scheduling, as we have done in Case I.

Power allocation in this case is also formulated as multiple-choice knapsack problem, as we have done in the previous case. The only difference is that, now we have more items in each item set: $(p_{n1}, w_{n1}), \cdots, (p_{nR}, w_{nR})$ in set S_n. Thus we should change the definition of S_n in Case I to

$$S_n = \{1, 2, \cdots, R\}.$$

Base service assignment, seed burst scheduling and MCKP power allocation constitute our new *Multiple-choice Knapsack Scheme* (MCKS) for resource allocation in Ka-band satellite systems. This scheme has the following advantages:

- *Fair*: Under any condition, every downlink spot will be guaranteed certain base service.
- *Efficient*: High profit is expected from the fully utilization of power and antennas in this scheme.
- *Fast*: MCKP is much easier than MCMKP or MKP and can be solved in pseudo-polynomial time. Also, in our particular application, the size of MCKP ($N \times R$) is much smaller than the size of MCMKP ($M \times L \times R$) or MKP ($M \times L$), which further reduces the computing time.

These advantages will be demonstrated by the simulation and computation results in the next chapter.

Chapter 5: Simulation and Results

In this chapter, we describe the OPENT models used to simulate the Ka-band satellite network scenarios and resource allocation schemes, and present the results of simulation. In particular, performance of our MCKP scheme is compared with that of Birmani's MKP scheme. Algorithms for solving the relevant knapsack problems are provided in Appendix A.

5.1 OPNET Simulation Model

Simulation is a very useful tool for performance evaluation of protocols or schemes in network systems. When the system to be characterized is still at the design stage, simulation provides an easy and quick way to predict a new scheme's performance or compare performances of several alternative schemes.

In this work, the OPNET simulator is selected to build the network simulation models. OPNET simulator is event-driven and operates at three hierarchical levels to describe and control the network to be analyzed. These are the network level, the node level and the process level. The network level consists of network nodes connecting each other by links. The node level comprises different function modules inside each network node, for example, traffic generator, packet queue and processor. The actual operations, algorithms or schemes are implemented in the process level.

Figure 5.1 shows the network level OPNET model used to simulate the system and resource allocation strategies discussed in previous chapters.

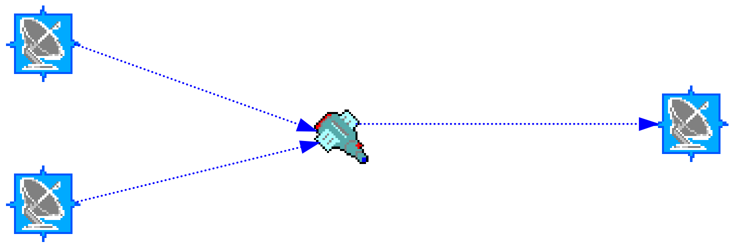


Figure 5.1: OPNET simulation model

The leftmost two nodes, *BurstData* and *WebLoad*, are responsible for generating various types of uplink traffic. The *Satellite* node simulates the functions of the satellite: receiving packets from uplinks, onboard switching and scheduling, and transmission of packets to downlinks. The resource allocation schemes designed for NOCC are also simulated onboard the satellite to make the simulation model easier. The *Downlink* node collects the packets and simulation results.

5.2 Traffic Models

Traffic modeling plays an important role in the design and simulation of communication networks [25]. Since the traffic models describe the statistical patterns of the information the objective network is expected to carry, accurate and practical traffic modeling is fundamental to successful network design and capacity planning. This is even more crucial for broadband satellite networks because the diversified types of applications that will access the satellite channels: from single home user to Internet backbone nodes, from very low data rates to gigabits per second, from bursts of a few milliseconds to long duration calls, from best-effort delivery to fully guaranteed high-priority data forwarding. Therefore, the traditional Poisson model is no longer applicable to our broadband satellite network simulation. Modern traffic models are required to make the results more meaningful and credible.

In this work, we consider two typical traffic models:

- Weibull-lognormal model for Web and bulk data transfer workload traffic;
- Markov-Modulated Poisson Process (MMPP) model for connectionless bursty data traffic;

5.2.1 Web and Bulk Data Transfer Workload

The World Wide Web (WWW) and bulk data transfer traffic constitutes the majority of the current Internet traffic volume. Barrett [26] has found that this type of traffic can be modeled at the connection level by fitting statistical distributions to two key traffic variables: connection interarrival times, and connection transmission ("download") sizes. Also he showed that Weibull distribution is the best fit for the interarrival time, while transmission size can be well characterized by the log-normal distribution.

Weibull distribution

Weibull distribution is a popular heavy-tailed distribution in network traffic modeling. Its probability density function is:

$$f(x) = (\frac{\beta}{\alpha})(\frac{x}{\alpha})^{\beta-1} e^{-(\frac{x}{\alpha})^{\beta}},\ \alpha, \beta > 0,\ x > 0.$$

And the distribution function is

$$F(x) = 1 - e^{-(\frac{x}{\alpha})^{\beta}}.$$

The Weibull samples can be generated in two steps:

(1) generate a sample uniformly distributed in (0,1), i.e. $u \sim U(0,1)$;

(2) obtain Weibull sample *x* using inverse of the Weibull distribution function:

$$x = \alpha[-\ln(1-u)]^{\frac{1}{\beta}}.$$

Log-normal distribution

Log-normal distribution is one of the early non-exponential distributions applied to network traffic modeling. The definition of the log-normal distribution is based on the normal distribution: given that $Y = \log(X)$ is normally distributed $Y \sim N(\mu, \sigma^2)$, the random variable *X* shall be called log-normal distributed. Its density function takes the form:

$$f(x) = \frac{1}{x\sqrt{2\pi}\sigma} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}, \ x > 0.$$

And its mean and variance are:

$$E[X] = e^{\mu + \frac{\sigma^2}{2}},$$

$$\operatorname{var}[X] = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1).$$

The log-normal samples can also be generated in two steps:

(1) generate a normally distributed sample $y \sim N(\mu, \sigma^2)$;

(2) obtain the log-normal sample *x* by transforming *y* via the relation:

$$x = e^y.$$

OPNET implementation

The connection level simulation is accomplished by dynamic process. At the beginning of simulation, a parent process is entered, and it will invoke a child process each time a new connection starts. The operations of parent and child processes are as follows:

- *Parent Process*: generate a Weibull distributed interarrival time *t*, wait for an amount of time *t* and create a new child process, generate another *t* and repeat the above operations until the simulation ends.
- *Child Process*: as soon as it is invoked, it randomly picks a log-normal transmission size and divides the total size into packets. It then sends out these packets and closes itself after that.

5.2.2 Connectionless Bursty Data

The connectionless bursty data type includes a large number of relatively less interactive traffic sources. It can be simulated by MMPP model [27].

MMPP model

A general *n*-state MMPP is completely determined by two matrices: the state transition rate matrix Λ and the arrival rate matrix *A*, as shown below [28]:

$$\Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1n} \\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{n1} & \vdots & \vdots & \vdots \end{bmatrix}, \qquad A = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots & a_{nn} \end{bmatrix}.$$

The corresponding Markov-modulated Poisson process has following characteristics [29]:

- The time T_i the underlying Markov chain will stay in state *i* is exponentially distributed with parameter $\frac{1}{\lambda_{ii}}$.
- In each state *i*, events arrive according to a Poisson process with rate a_i.

- As the time T_i expires, the underlying Markov chain jumps from state *i* to state *j* ($i \neq j$), with probability p_{ij} given by

$$p_{ij} = \frac{\lambda_{ij}}{\sum_{k \neq i} \lambda_{ik}} \ .$$

Let $P = (P_1, \cdots, P_n)$ denotes the vector of the steady-state probabilities for the underlying Markov chain, it should satisfy the following equations:

$$P\Lambda = P \qquad \text{and} \qquad P_1 + P_2 + \cdots + P_n = 1.$$

And the mean arrival rate of MMPP is given by

$$\lambda = \sum_{i=1}^{n} P_i a_i \ .$$

OPNET implementation

A two-state MMPP model is implemented in our simulation as in Figure 5.2.

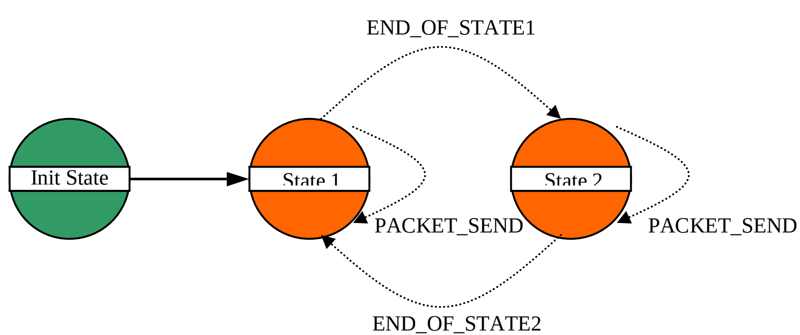


Figure 5.2: Two-state MMPP OPNET model

The green color stands for forced state, which means that the visit to this state can only be transitory, while the orange color stands for unforced state. The operations of these three states are described below:

- *Init State*: This state is only entered once at the beginning of simulation. It sets parameters and initializes variables, then goes into State 1.
- *State 1*: There are three types of events in this state: (1) if this is the new entrance of State 1, determine the exponential end time T_1 of State 1. (2) generate an exponential interarrival time *t*, if *t* is earlier than T_1, schedule the next packet generation. (3) if T_1 expires, transfer to State 2.
- *State 2*: This state's operation is similar to State 1 with different distribution parameters.

5.3 Resource Allocation Schemes

Figure 5.3 shows the onboard processor model in Satellite node. The state *arrival* is responsible for queuing received packets in downlink buffers according to their destination addresses. Each time the satellite receives *Weather Change* message, it will update its information in state *weather*. The resource allocation schemes are implemented in state *burst,* where they are called when the next burst is coming.

A polynomial-time approximate algorithm MTHM proposed by Martello and Toth [23] was used in Birmani's work [4]. But computational experiments [12] indicated that MTHM works better for uncorrelated items and dissimilar capacities, which is not case in our problem.

Pisinger presents a new exact algorithm MULKNAP for MKP in [14], which is specially designed for solving large problem instances. This MULKNAP algorithm is much faster than the MTHM approximation algorithm, so it is used in our simulation as an alternative to MTHM to solve the MKP.

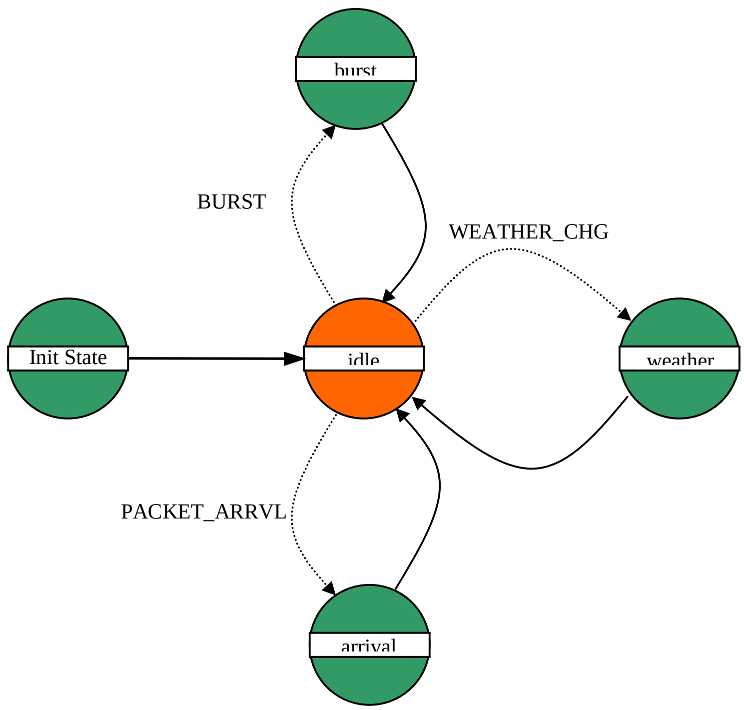


Figure 5.3: Satellite onboard processor OPNET model

The algorithm MCKNAP we used in our work to solve MCKP was also proposed by Pisinger [30]. MCKP is *NP-hard*, but it is much easier compared to MKP, which is *NP-hard* in the strong sense. As a result, the MCKP algorithm is much faster than the MKP algorithms.

Pseudo-codes of these three algorithms are listed in Appendix A. Table 5.1 below compares their computing times for same data instances. We can see that the approximate MKP algorithm MTHM is much less efficient than the exact MKP algorithm MULKNAP. Therefore we adopt MULKNAP for solving MKP, and MCKNAP for solving MCKP respectively in the OPNET simulation.

Table 5.1: Computing time in seconds of three algorithms

Rain Area	0	2%	5%	8%	10%	12%	15%	18%
MTHM	13.170	12.899	13.670	13.770	13.459	13.738	13.677	13.840
MULKNAP	0.110	0.281	0.279	0.317	0.289	0.309	0.277	0.292
MCKNAP	0.012	0.010	0.012	0.013	0.012	0.011	0.012	0.012

5.4 Simulation Results

In this section, we compare the performances of our MCKP approach and Birmani's MKP approach in Ka-band resource allocation.

Eight different scenarios in terms of rain fade area are simulated: no rain, rain fade area 2%, 5%, 8%, 10%, 12%, 15% and 18%. 10% rain area is the critical case, beyond which we cannot guarantee standard rate service to every downlink. Since fade in heavy rain is more severe than that in light rain, one downlink in heavy rain will be counted as two downlinks in light rain when we consider the extra power entailed by the rain fade. To be precise, when we say "rain fade area", we always mean the "equivalent" fade area due to light rain. In our implementation, when a rain fade area, say 5%, is given, the distribution of light rain spots and heavy rain spots will be generated randomly.

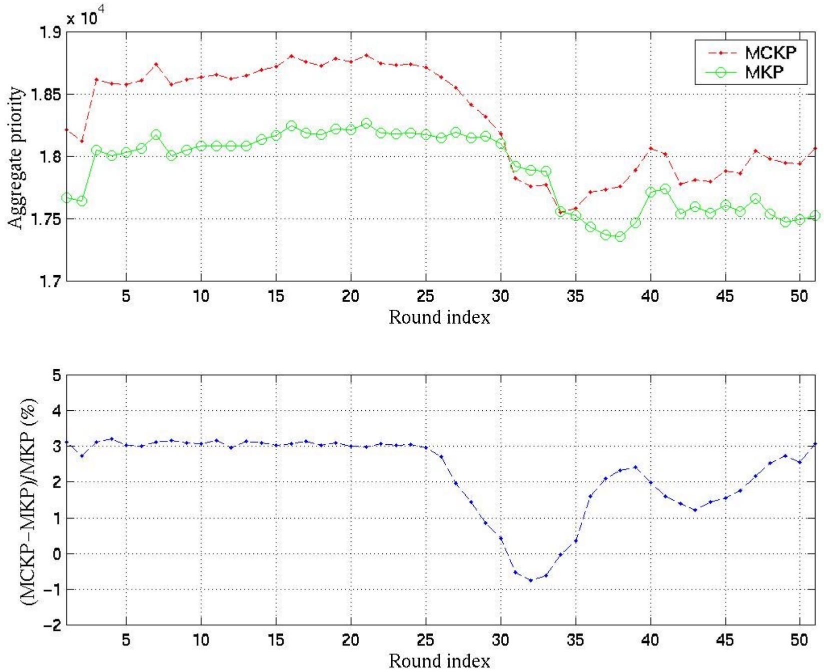
In the simulation, we have 700 downlinks and 20 antennas, thus each round consists of 35 bursts. For each scenario, we run 50 rounds. To compare the MCKP scheme and the MKP scheme, exactly the same traffic flow is fed into the MCKP algorithm and the MKP algorithm, and the two schemes will run independently. We will compare the aggregate priorities gained in each round and call them *dynamic* results. We use the term "dynamic" in consideration of the fact that, under each scheme, the system behaves continuously like a dynamic system, and we are comparing the long-term behaviors of these two systems. It should be noted that, if we look at the data for both schemes in a particular round, they might be different for all rounds except for the first one.

The *static* results are also furnished to provide a different view of comparison. The static results are obtained by applying both schemes to exactly the same traffic data in *every round*. To be specific, we run the system under the MCKP scheme continuously for 50 rounds; at the beginning of each round, the current data in downlink buffers is fed into the MKP scheme. The meaning of "static" is now evident: the time horizon for performance comparison is a *round*, as opposed to a long time period in the dynamic case. The computation of static results was carried out outside the OPNET environment although the data was taken from the OPNET simulation. All the computation results reported in the thesis were done in SUN Ultra10 workstations.

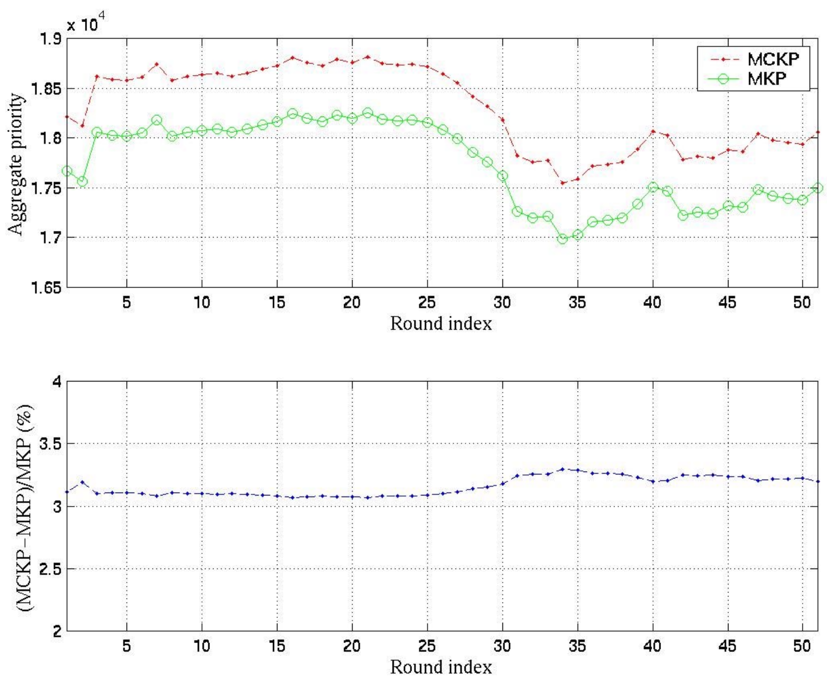
In the following subsections, we will compare the two schemes in terms of the aggregate priority in each round, computation time, power utilization, antenna utilization, and service missing for downlinks in each of the scenarios. These results are closely related to the performance metrics we discussed in Section 4.3.

5.4.1 Aggregate Priority

Figures 5.4-5.11 show the aggregate priority comparison between the MCKP and MKP schemes under various rain conditions. In each figure, the first two plots illustrate the dynamic simulation results, while the other two show the static simulation results.

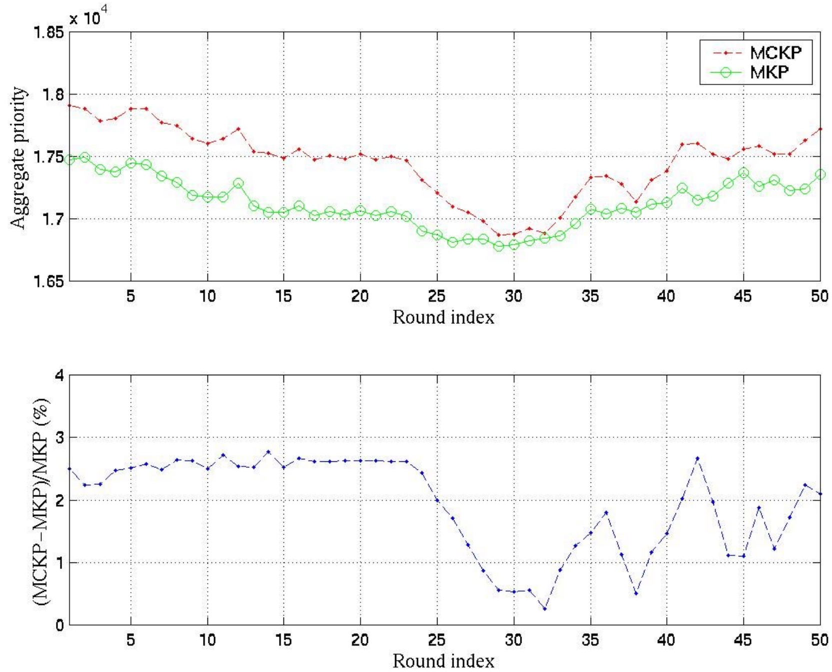


(a) Dynamic results

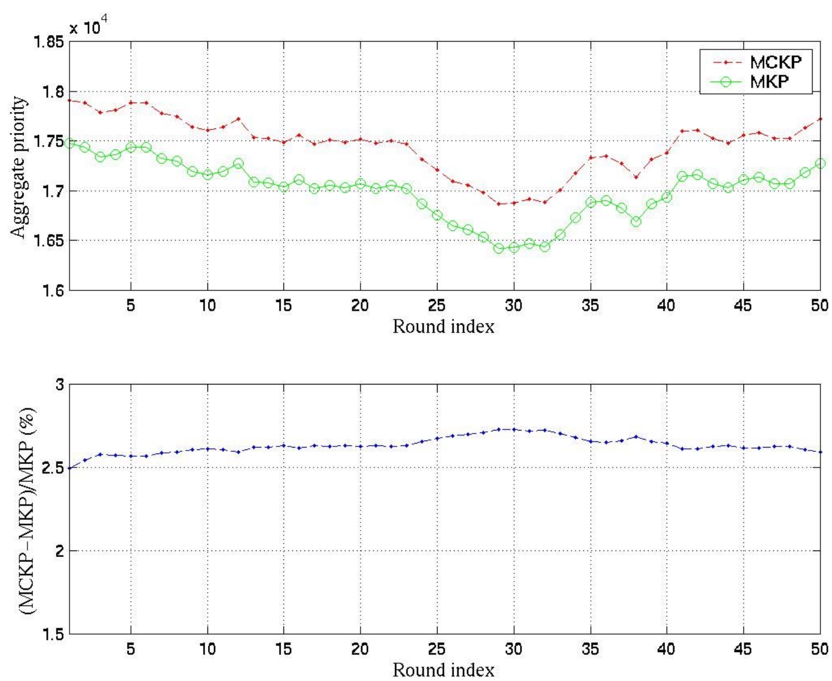


(b) Static results

Figure 5.4: Aggregate priorities under no rain condition

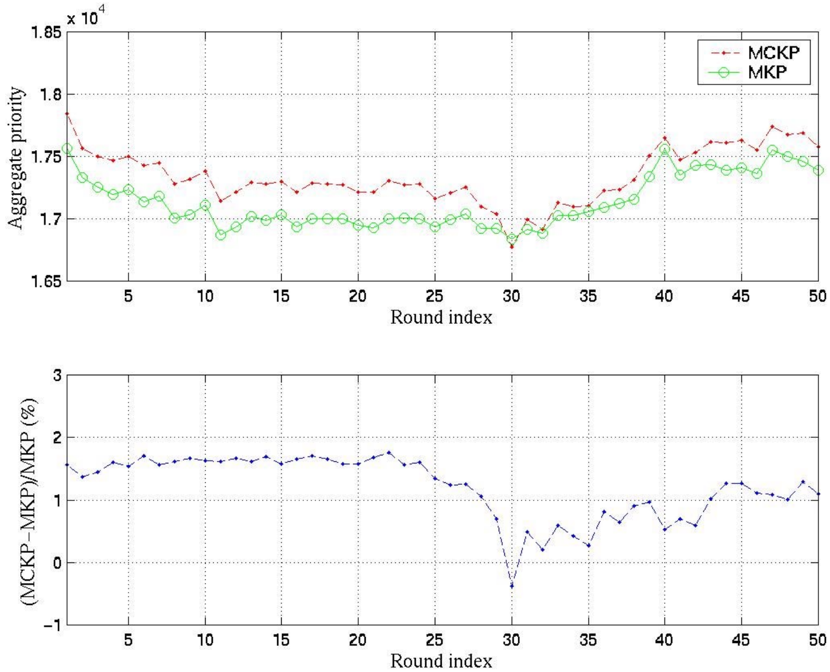


(a) Dynamic results

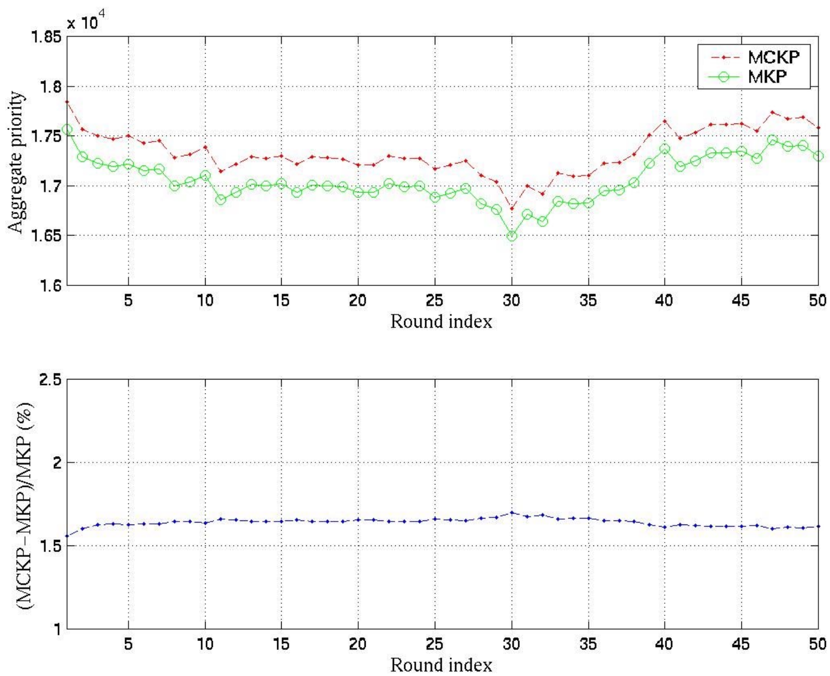


(b) Static results

Figure 5.5: Aggregate priorities with rain fade area 2%

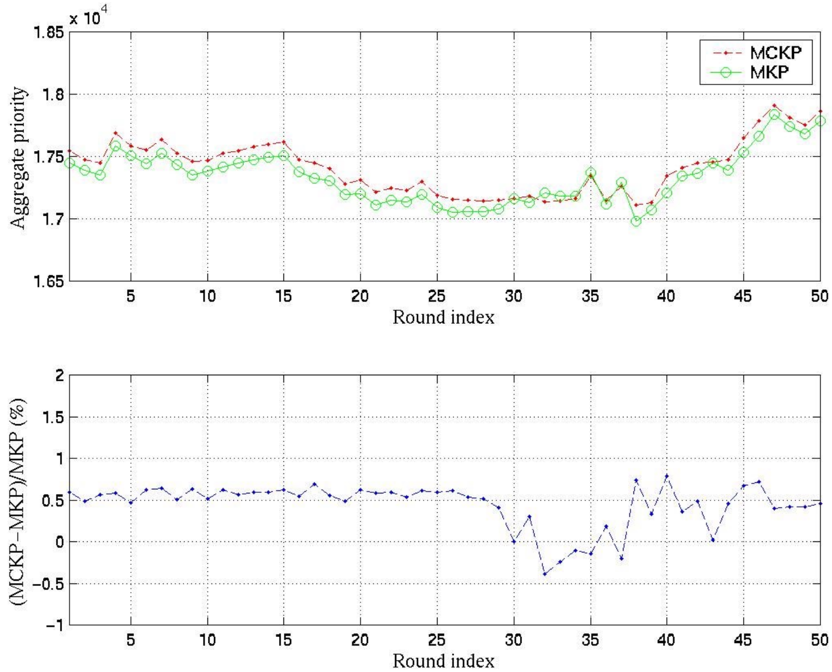


(a) Dynamic results

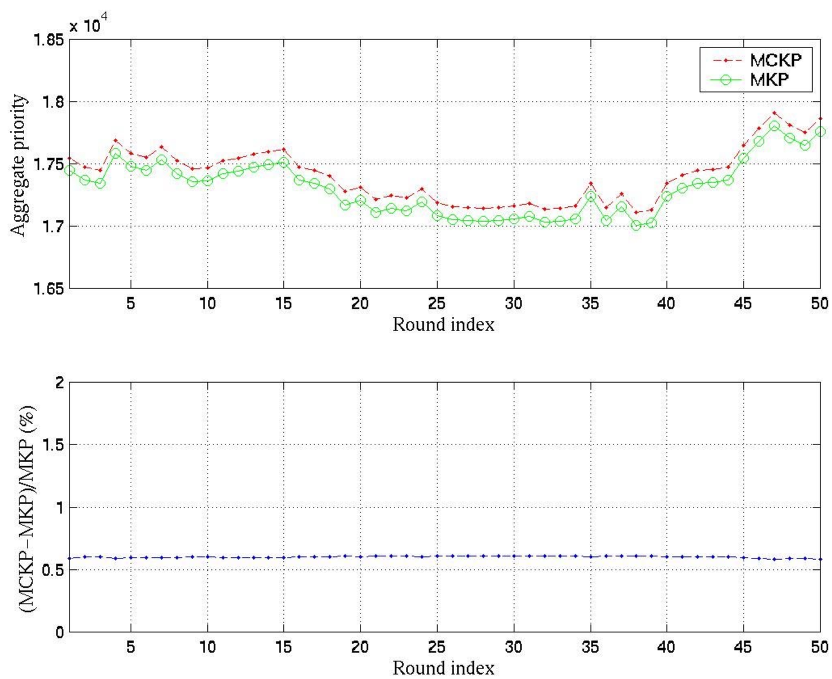


(b) Static results

Figure 5.6: Aggregate priority with rain fade area 5%

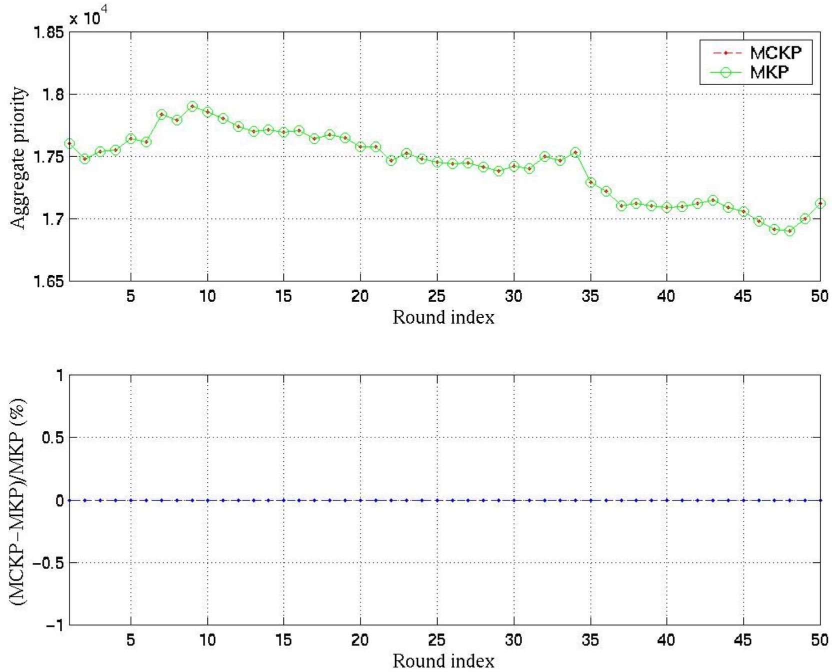


(a) Dynamic results

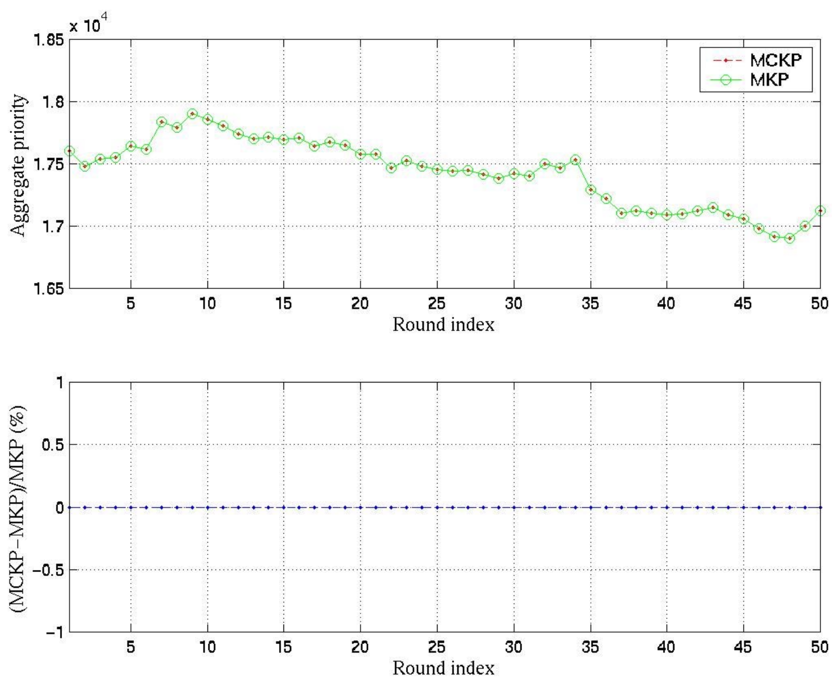


(b) Static results

Figure 5.7: Aggregate priority with rain fade area 8%

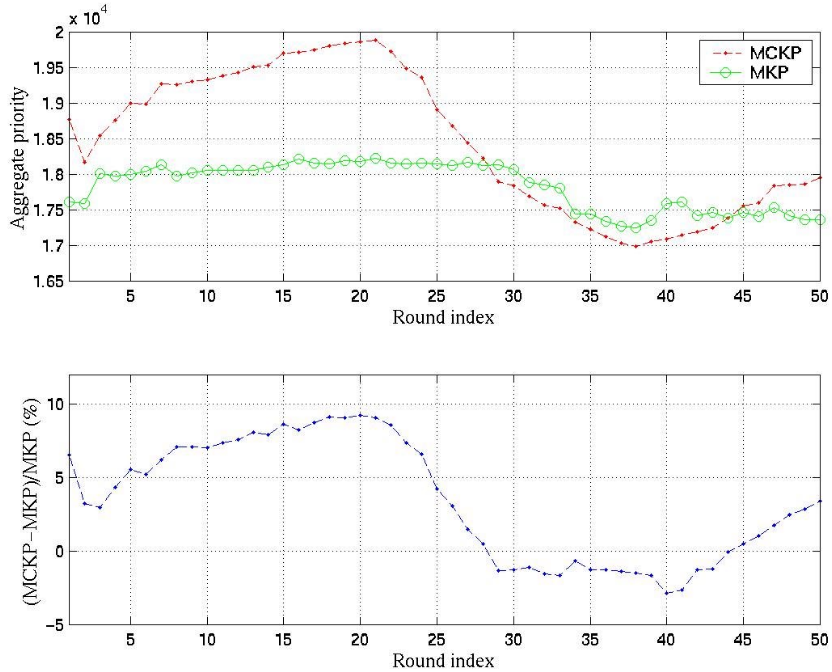


(a) Dynamic results

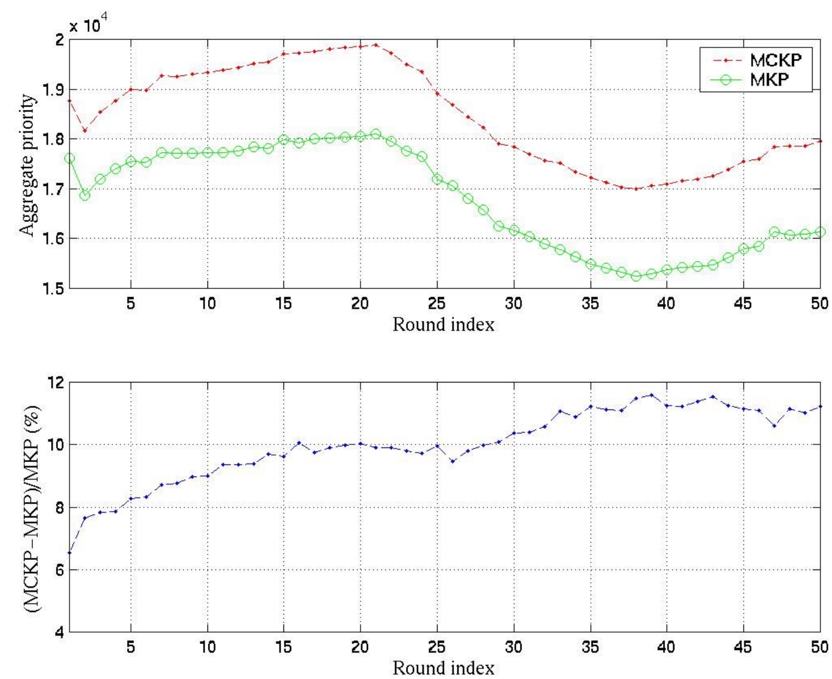


(b) Static results

Figure 5.8: Aggregate priority with rain fade area 10%

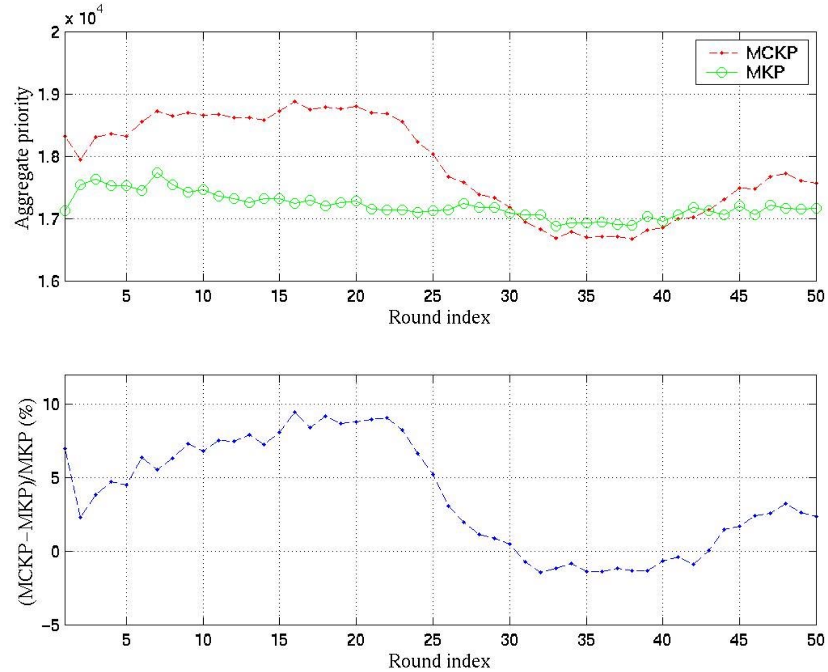


(a) Dynamic results

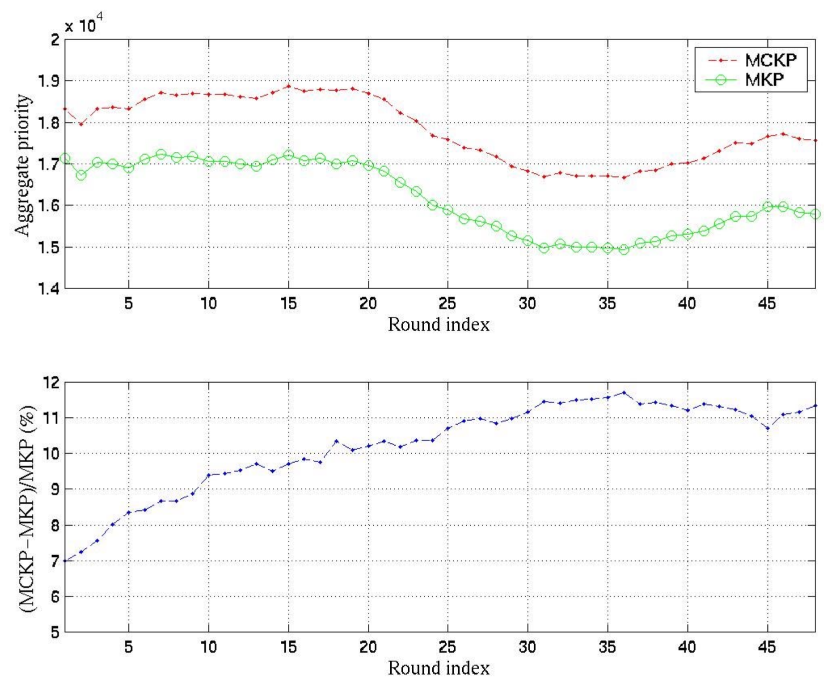


(b) Static results

Figure 5.9: Aggregate priority with rain fade area 12%

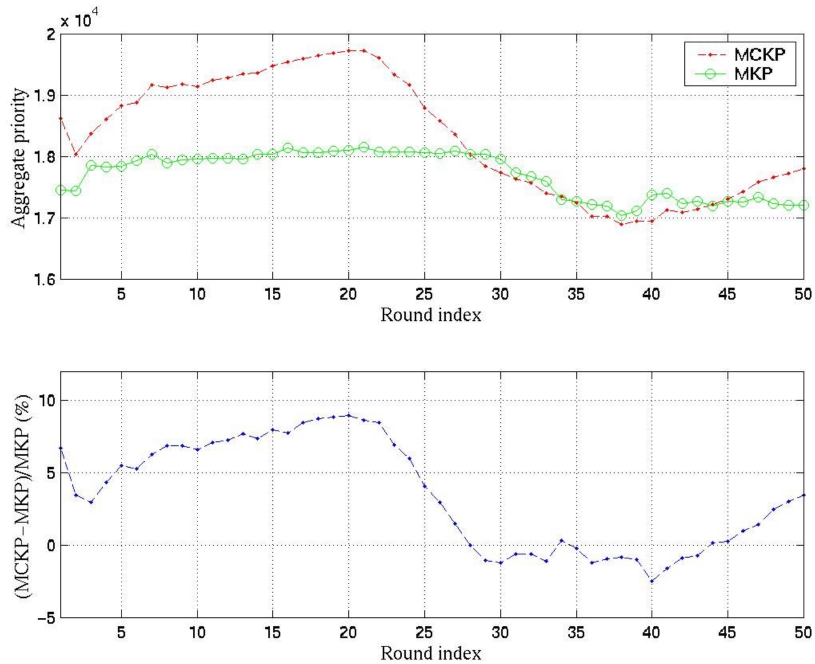


(a) Dynamic results

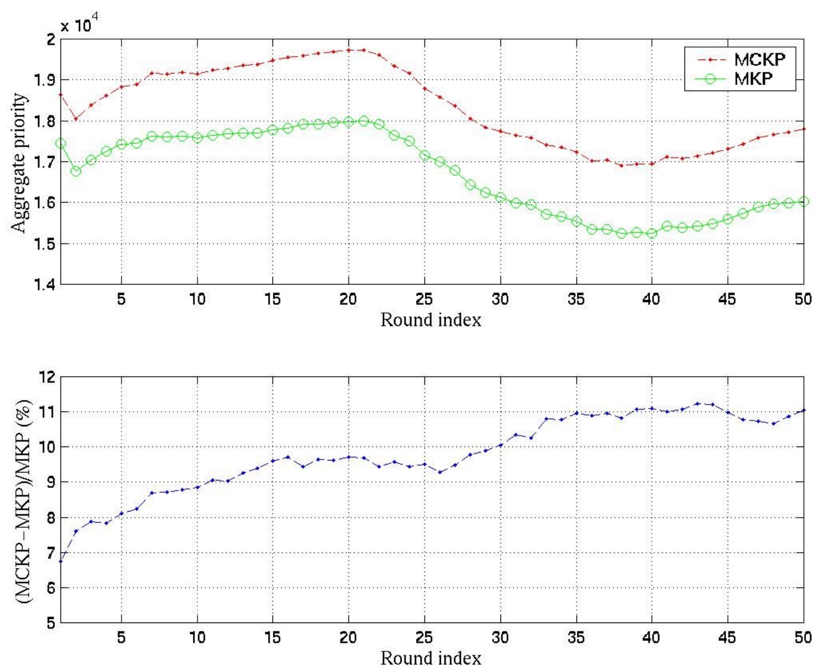


(b) Static results

Figure 5.10: Aggregate priority with rain fade area 15%



(a) Dynamic results



(b) Static results

Figure 5.11: Aggregate priority under 18% rain condition

By averaging the advantage of MCKP over MKP in all rounds for each scenario, we obtain Figure 12, which reflects the information contained in Figures 4-11 in a compact way. The following observations can be made:

- MCKP outperforms MKP in both dynamic case and static case.
- The advantage of MCKP over MKP decreases with rain fade area when rain fade area is below 10%, vanishes at 10%, starts to increase beyond 10% and gets saturated around 15%. This can be explained as follows: when there is no rain, MKP wastes all the extra power while MCKP fully utilizes the power all the time; with increase of the rain fade area, MKP begins to make use of more and more extra power; when rain fade area reaches 10%, both MKP and MCKP use up the extra power and their performances coincide; beyond 10%, MKP drops some downlinks to save power for higher priority ones and the power will not be used exactly in full until the rain fade get much worse like around 15%.
- In each scenario, the patterns for trajectories of MCKP and MKP are similar in the static case, while they can be quite different in the dynamic case. This follows from the different ways of obtaining dynamic and static results, as discussed earlier.

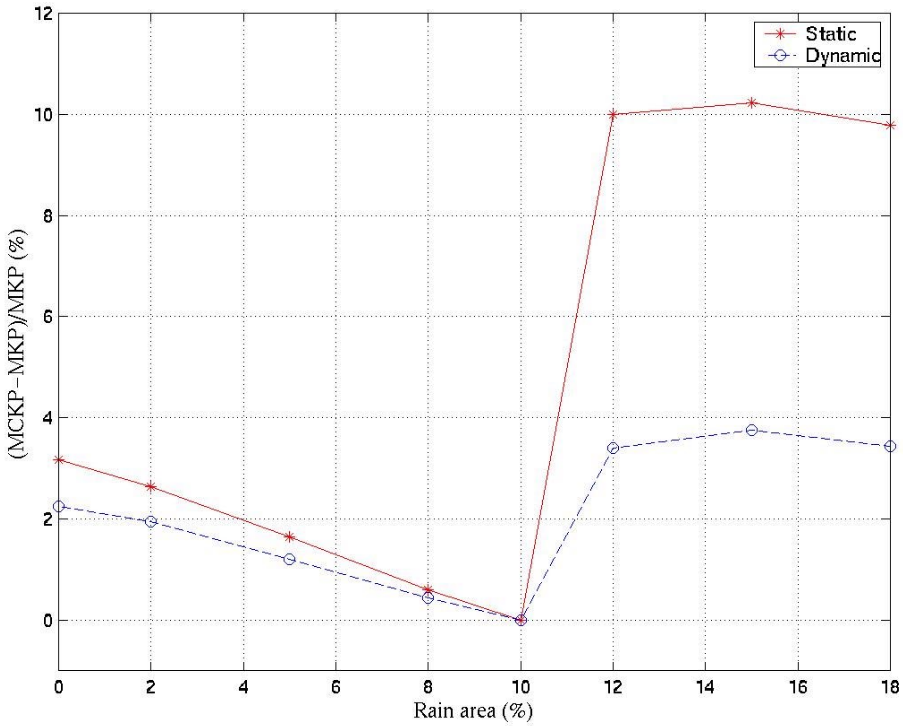


Figure 5.12: Advantage of MCKP over MCP vs. rain area

5.4.2 Computing Time

Table 5.2 gives the average computing time of MCKP and MKP algorithms under various rain conditions. In the MCKP scheme, all downlinks are first ordered according to their average priorities, when the *Sorting* time is taken, then each burst is considered as a knapsack and MCKNAP is called to solve it in a *Single MCKP* time. The total 35 bursts in one round add up to the *35 MCKP* time. Summing *Sorting* and *35 MCKP* results in the *Total* time.

As we analyzed in previous chapters, MCKP is easier than MKP in itself, and in our application, the MCKP has much smaller size than the MKP. These factors lead to its much faster computation speed than that of MKP, as one can see from Table 5.2.

Table 5.2: Computing times of MCKP and MKP

Rain Area	MCKP (ms)				MKP (ms)
	Sorting	Single MCKP	35 MCKP	Total	
0	0.61	0.34	11.84	12.45	109.80
2%	0.62	0.36	12.58	13.20	281.40
5%	0.61	0.32	11.33	11.94	279.39
8%	0.65	0.33	11.44	12.09	283.01
10%	0.62	0.33	11.47	12.09	287.06
12%	0.60	0.33	11.66	12.26	317.87
15%	0.61	0.35	12.26	12.87	631.56
18%	0.63	0.35	12.33	12.96	275.61

5.4.3 Resource Utilization

Table 5.3 shows the utilization of power and antenna for both schemes. The MCKP scheme utilizes almost all the available resources under any weather condition, as one would expect from the design of the scheme. The MKP scheme fails to make full use of the resources (either power or antenna) except when rain fade area is 10%, as we analyzed in Subsection 4.3.2.

Table 5.3: Resource utilization and service missing

Rain Area	Power Utilization		Antenna Utilization		Service Missing	
	MCKP	MKP	MCKP	MKP	MCKP	MKP
0	99.8%	96.6%	100%	100%	No	No
2%	99.8%	97.3%	100%	100%	No	No
5%	99.8%	98.2%	100%	100%	No	No
8%	99.8%	99.2%	100%	100%	No	No
10%	99.8%	99.4%	100%	100%	No	No
12%	99.8%	99.4%	100%	99.6%	No	Yes
15%	99.8%	99.5%	100%	98.9%	No	Yes
18%	99.8%	99.4%	100%	98.3%	No	Yes

5.4.4 Service Missing

From Table 5.3, we can also see that, with the MCKP scheme, every downlink will get service in any round; while with the MKP scheme, service missing occurs when rain fade area is greater than 10%. This conflicts with our criterion of fairness. Figures 5.13-5.16 give a closer view of the service missing situation with the MKP scheme. We collect the data in 50 rounds.

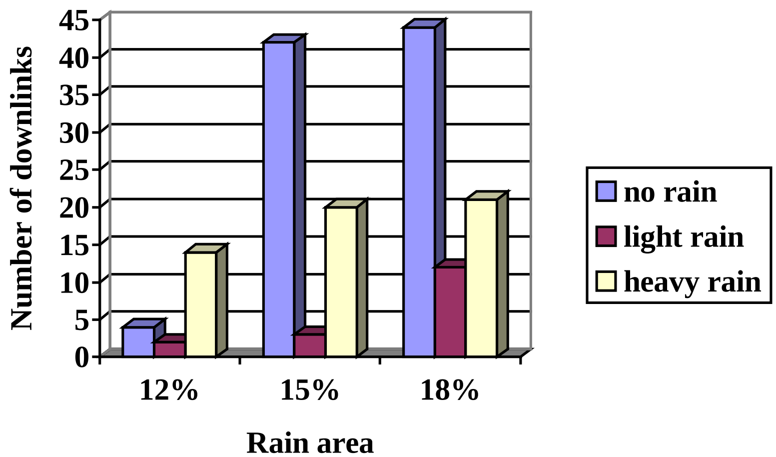


Figure 5.13: Number of out-of-service downlinks vs. rain area

Figure 5.13 shows the number of downlinks who were out of service at least once in 50 rounds under several rain conditions. The *no rain, light rain, heavy rain* legends denote the downlinks that were suffering no rain fade, light rain fade and heavy rain fade, respectively. It can be seen that the total number of downlinks being missed at least once increases with the rain area.

In Figures 5.14, 5.15 and 5.16, we show the number of downlinks that experienced various extents of service missing under different rain conditions. We notice that when the rain area is 18%, about 10 downlinks got no service in more than 25 rounds out of 50 rounds, which is unacceptable in many applications.

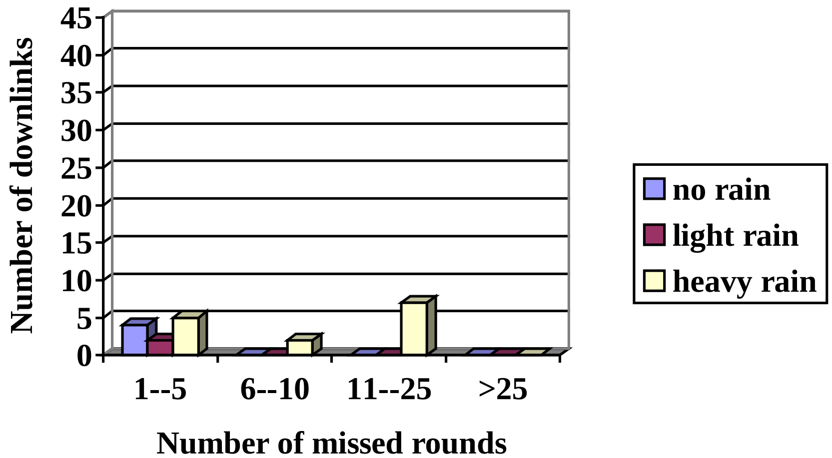


Figure 5.14: Histogram of number of missed rounds (12% rain area)

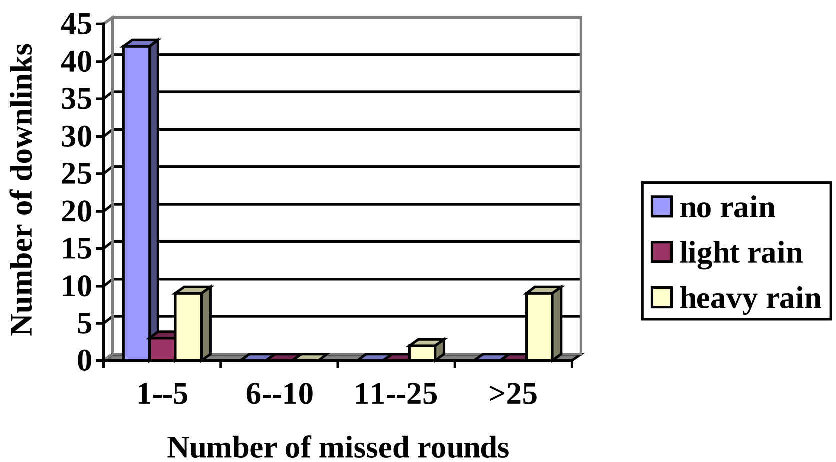


Figure 5.15: Histogram of number of missed rounds (15% rain area)

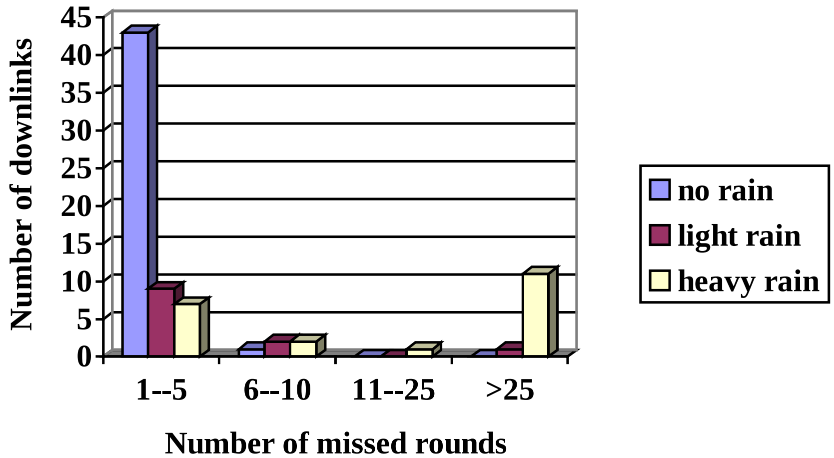


Figure 5.16: Histogram of number of missed rounds (18% rain area)

Chapter 6: Conclusions and Future Work

6.1 Conclusions

With the fast growth of the Internet, the Ka-band satellite system has gained more and more interests for its huge bandwidth in recent years. Rain fading presents one of the most challenging problems in utilization of Ka band. Several compensation approaches were proposed in the literature. In this thesis, we have presented a new compensation mechanism for downlink transmission through the dynamic resource allocation.

The resources considered include power and antennas on board the satellite. We mathematically formulate the problem in the framework of the Knapsack Problems. The objective is to maximize the aggregate priority of transmitted packets as well as maintain the fairness among downlinks. We have shown that the coupled power and resource allocation problem can be described as a Multi-choice Multiple Knapsack Problem (MCMKP), which is unlikely to solve in reasonable time.

To make the original problem tractable, we decouple the power allocation and the burst scheduling problems. First the bursts are scheduled using the seeding theory, which divides the “multiple-knapsack” into separate “single knapsacks”. Then power allocation in each knapsack is modeled as a Multiple-choice Knapsack Problem (MCKP), which is much easier to solve than MCMKP.

The Multiple Knapsack Problem (MKP) approach proposed by Birmani is also introduced in this thesis for comparison. Performances of these two schemes are demonstrated through OPNET simulations. Comparisons of aggregate profit, computing time and resource utilization have shown that the MCKP scheme outperforms the MKP scheme in terms of fairness, efficiency and computation speed.

6.2 Future Work

Future work in this area includes more practical and quantitative metrics which might be more closely related to the industrial interest. Also, more specific traffic models for the interested system can be used in designing allocation schemes.

Uplink rain compensation approaches, such as Uplink Power Control (UPC), can be integrated with the downlink rain compensation approaches to obtain more efficient resource allocation and rain fade mitigation schemes.

Joint consideration of coding rate change, power control and transmission rate adjustment is another possible and interesting research direction.

Other valuable approaches may include the parallel computing of the KP algorithms, joint consideration of congestion control on board the satellite and accurate rain fade prediction in NOCC.

Appendix A: Algorithms for Solving Knapsack Problems

A.1 Approximation Algorithm for MKP

A polynomial-time approximate algorithm proposed by Martello and Toth [23] was used in Vineet's approach. It works as follows. The items are first sorted so that

$$\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots \geq \frac{p_n}{w_n} \quad . \tag{A.1}$$

Then the following procedure MTHM applies.

procedure MTHM:

input: $n, m, (p_j), (w_j), (c_i)$ **;**

output: $z, (y_j)$ **;**

begin

1. [initial solution]

   $z = 0$;

   for $j = 1$ *to* n **do** $y_j = 0$ **;**

   for $i = 1$ *to* m **do**

   begin

   $\bar{c}_i = c_i$ **;**

   call *GREEDYS;*

   end;

2. [rearrangement]

$z = 0$;

for $i = 1$ *to* m **do** $\bar{c}_i = c_i$**;**

$i = 1$;

for $j = n$ *to* 1 **do if** $y_j > 0$ **then**

begin

let l be the first index in $\{i, \ldots, m\} \cup \{1, \ldots, i-1\}$ *such that* $w_j \leq \bar{c}_l$ *;*

if no such *l* **then** $y_j = 0$ **else**

begin

$y_j = l$ **;**

$\bar{c}_l = \bar{c}_l - w_j$ **;**

$z = z + p_j$ **;**

if $l < m$ **then** $i = l + 1$ **else** $i = 1$

end

end

for $i = 1$ *to* m **do call** *GREEDYS;*

3. [first improvement]

for $j = 1$ *to* n **do if** $y_j > 0$ **then**

for $k = j + 1$ *to* n **do if** $0 < y_k \neq y_j$ **then**

begin

$h = \arg\max\{w_j, w_k\}$ **;**

$l = \arg\min\{w_j, w_k\}$ **;**

$d = w_h - w_l$ **;**

if $d \le \bar{c}_{y_l}$ *and* $\bar{c}_{y_h} + d \ge \min\{w_u : y_u = 0\}$ **then**

begin

$t = \arg\max\{p_u : y_u = 0 \text{ } and \text{ } w_u \le \bar{c}_{y_h} + d\}$ **;**

$\bar{c}_{y_h} = \bar{c}_{y_h} + d - w_t$ **;**

$\bar{c}_{y_l} = \bar{c}_{y_l} - d$ **;**

$y_t = y_h$ **;**

$y_h = y_l$ **;**

$y_l = y_t$ **;**

$z = z + p_t$

end

end

4. [second improvement]

for $j = n \text{ } to \text{ } 1$ **do if** $y_j > 0$ **then**

begin

$\bar{c} = \bar{c}_{y_j} + w_j$ **;**

$Y = \Phi$ **;**

for $k = 1 \text{ } to \text{ } n$ **do**

if $y_k = 0 \text{ } and \text{ } w_k \le \bar{c}$ **then**

begin

$Y = Y \cup \{k\}$ **;**

$\bar{c} = \bar{c} - w_k$ **;**

end

if $\sum_{k \in Y} p_k > p_j$ **then**

begin

for each $k \in Y$ **do** $y_k = y_j$ **;**

$\bar{c}_{y_j} = \bar{c}$ **;**

$y_j = 0$ **;**

$z = z + \sum_{k \in Y} p_k - p_j$

end

end

end

procedure GREEDYS:

input: $n, (p_j), (w_j), z, (y_j), i, \bar{c}_i$ **;**

output: $z, (y_j)$ **;**

begin

for $j = 1\ to\ n$ **do**

if $y_j = 0\ and\ w_j \le \bar{c}_i$ **then**

begin

$y_j = i$ **;**

$\bar{c}_i = \bar{c}_i - w_j$ **;**

$z = z + p_j$ **;**

end

end.

A.2 Exact algorithm for MKP

Pisinger presents a new exact algorithm for MKP in [14], which is specially designed for solving large problem instances. The main algorithm MULKNAP and the recursive branch and bound algorithm are briefly described as follows.

procedure MULKNAP(n, m, p, w, x, c):

Order the capacities $c_1 \leq c_2 \leq \cdots \leq c_m$,

for $j = 1$ *to* n **do**

$d_j = 1$ **;**

for $i = 1$ *to* m **do** $x_{ij} = 0$ **;** $y_{ij} = 0$ **; rof;**

rof;

$z = 0$ ***;***

MULBRANCH (0,0,0, $c_1, \cdots, c_m$ *);*

procedure MULBRANCH ($h, P, W, c_1, \cdots, c_m$):

Tighten the capacities c_i *by solving* m *Subset-sum problems defined on* $h+1, \cdots, n$.

Solve the surrogate relaxed problem with capacity $c = \sum_{i=1}^{m} c_i$ *. Let* x' *be the solutionot this problem, with objective value u.*

if $(P + u > z)$ **then**

split the solution x' *in the m knapsacks by solving a series of Subset-sum problems defined on items with* $x_j' = 1$. *Let* y_{ij} *be the optimal filling of* c_i *with corresponding profit sum* z_i.

Improve the heuristic solution by greedy filling knapsacks with $\sum_{j=h+1}^{n} w_j y_{ij} < c_i$.

if $(P + \sum_{i=1}^{m} z_i > z)$ **then** *copy y to x, set* $z = P + \sum_{i=1}^{m} z_i$. **fi**;

fi;

if $(P + u > z)$ **then**

reduce the items by using some upper bound tests, and sway the reduced items to the first positions, increasing h.

let I be the smallest knapsack with $c_i > 0$. *solve an ordinary 0-1 knapsack problem with* $c = c_i$ *defined on the free variables. The solution vector is* x'*. Choose the branching item l as the item with largest profit-to-weight ratio among items* $x_j' = 1$.

Swap l to position $h+1$ *and set* $j = h+1$.

Let $y_{ij} = 1$***;*** *{assign item j to knapsack i}*

MULBRANCH ($h+1, P + p_j, W + w_j, c_1, \cdots, c_i - w_j, \cdots, c_m$ *);*

Let $y_{ij} = 0$***;*** *{exclude item j from knapsack i}*

Set $d' = d_j$*;* $d_j = i+1$*;*

MULBRANCH ($h, P, W, c_1, \cdots, c_m$ *);*

Find j again, and set $d_j = d'$.

fi;

A.3 Exact Algorithm for MCKP

The algorithm we used in our work to solve MCKP was also proposed by Pisinger [30]. The main algorithm MCKNAP and partitioning algorithm PARTITION are sketched here.

procedure MCKANP:

Solve LMCKP through the partitioning algorithm.

Determine gradients $L^{+} = \{\lambda_i^{+}\}$ *and* $L^{-} = \{\lambda_i^{-}\}$ *for* $i = 1, \cdots, k, i \neq a$.

Partially sort L^{+} *in decreasing order and* L^{-} *in increasing order.*

$z = 0$; $s = 1$; $t = 1$; $C = \{N_a\}$; $Y_C = reduceclass(N_a)$;

repeat

reduceset(Y_C); **if** ($Y_C = \Phi$) **then break**; **fi**;

$N_i = L_s^{-}$; $s = s + 1$;   {*choose next class from* L^{-} }

if (N_i *is not used*) **then**

$R_i = reduceclass(N_i)$;

if ($|R_i| > 1$) **then** *add*(Y_C, R_i);

fi;

forever;

Find the solution vector.

procedure PARTITION:

Step 0. Preprocess. *For all classes* $i = 1, \cdots, k$ *let* α_i *and* β_i *be indices to the items having minimal weight (resp. maximal profit) in* N_i. *In case of several items satisfying the criterion, choose te item having largest profit for* α_i *and smallest weight for* β_i. *Set W=P=0, and remove those items* $j \neq \beta_i$ *which have* $w_{ij} \geq w_{i\beta_i}$ *and* $p_{ij} \leq p_{i\beta_i}$, *since these are dominated by item* β_i. *If the class* N_i *has only one item left, save the LP-optimal choice* $b_i = \beta_i$ *and set* $W = W + w_{ib_i}$, $P = P + p_{ib_i}$, *then delete class* N_i.

Step 1. Choose median. *For M randomly chosen classes* N_i *define the corresponding slope* $\lambda_i = (\delta p_i / \delta w_i) = (p_{i\beta_i} - p_{i\alpha_i})/(w_{i\beta_i} - w_{i\alpha_i})$. *Let* $\lambda = (\delta \overline{p} / \delta \overline{w})$ *be the median of these M slopes.*

Step 2. Find the conclusion. *For each class* N_i *find the items which maximize the projection on the normal to* $(\delta \overline{w}, \delta \overline{p})$, *i.e. which maximize the determinant* $\det(w_{ij}, p_{ij}, \delta \overline{w}, \delta \overline{p}) = w_{ij} \delta \overline{p} - p_{ij} \delta \overline{w}$. *We swap these items to the beginning of the list such that they have indices* $\{1, \cdots, l_i\}$ *in class* N_i.

Step 3. Determine weight sum of conclusion. *Let* g_i, h_i *be the lightest (resp. heaviest) item among* $\{1, \cdots, l_i\}$ *in class* N_i, *and let* W' *and* W'' *be the corresponding weight sums. Thus* $W' = W + \sum_{i=1}^{k} w_{ig_i}$ *and* $W'' = W + \sum_{i=1}^{k} w_{ih_i}$.

Step 4. Check for optimal partitioning. *If* $W^{'} \leq c \leq W^{''}$ *the partitioning at* $(\delta\overline{w}, \delta\overline{p})$ *is optimal. First, choose the lightest items from each class by setting* $b_i = g_i, W = W + w_{ib_i}, P = P + p_{ib_i}$. *Then while* $W - w_{ig_i} + w_{ih_i} \leq c$ *run through the classes where* $l_i \neq 1$ *and choose the heaviest item by setting* $b_i = h_i$, $W = W - w_{ig_i} + w_{ib_i}$, $P = P - p_{ig_i} + p_{ib_i}$. *The first class where* $W - w_{ig_i} + w_{ih_i} > c$ *is the fractional class* N_a *and an optimal objective value to LMCKP is* $z_{LMCKP} = P + (c - W)\lambda$. *If no fractional class is defined, the LP-solution is also the optimal IP-solution. Stop.*

Step 5. Partition. *We have one of the following two cases: 1) if* $W^{'} > c$ *then the slope* λ *was too small. For each class* N_i *choose* β_i *as the lightest item in* $\{1, \cdots, l_i\}$ *and delete items* $j \neq \beta_i$ *with* $w_{ij} \geq w_{i\beta_i}$*; 2) If* $W^{''} < c$ *then the slope* $\lambda = (\delta\overline{p} / \delta\overline{w})$ *was too large. For each class* N_i *choose* α_i *as the heaviest item in* $\{1, \cdots, l_i\}$ *and delete items* $j \neq \alpha_i$ *with* $p_{ij} \leq p_{i\alpha_i}$ *(item j with* $w_{ij} \leq w_{i\alpha_i}$ *are too light, and items with* $w_{ij} > w_{i\alpha_i}$, $p_{ij} \leq p_{i\alpha_i}$ *are dominated). If the class* N_i *has only one item left, save the LP-optimal choice* $b_i = \beta_i$ *and set* $W = W + w_{ib_i}$, $P = P + p_{ib_i}$*, then delete class* N_i*. Goto Step 1.*

BIBLIOGRAPHY

[1] F. Gargione, T. Iida, F. Valdoni, and F. Vatalaro, "Services, technologies, and systems at Ka band and beyond – a survey", *IEEE J. Select. Areas Commun.*, vol. 17, no. 2, pp. 133-144, February 1999

[2] R. J. Acosta, R. Bauer, R. J. Krawczyk, R. C. Reinhart, M. J. Zernic, and F. Gargione, "Advanced communications technology satellite (ACTS): four-year system performance", *IEEE J. Select. Areas Commun.*, vol. 17, no. 2, pp. 193-203, February 1999

[3] E. J. Fitzpatrick, "SPACEWAY system summary", *Space Communications* , vol. 13, pp. 7-23, 1995

[4] V. Birmani, *Resource Allocation for Ka-band Broadband Satellite Systems*, Master Thesis, University of Maryland, 1999, http://www.isr.umd.edu/TechReports/CSHCN/1999/CSHCN_MS_99-6/CSHCN_MS_99-6.phtml

[5] R. J. Acosta, "Rain fade compensation alternatives for Ka band communication satellites", *NASA TM 107534*, 1996

[6] D. P. Connors, B. Ryu, and S. Dao, "Modeling and simulation of broadband satellite networks Part I: medium Access control for QoS provisioning", *IEEE Commun. Mag.*, pp. 72-79, March 1999

[7] J. Gilderson and J. Cherkaoui, "Onboard switching for ATM via satellite", *IEEE Commun., Mag.*, pp. 66-70, July 1997

[8] C. B. Cox, and T. A. Coney, "Advanced communications technology satellite adaptive rain fade compensation protocol performance", *NASA Technical Report*, 1996

[9] L. A. Wolsey, *Integer Programming*, Wiley-Interscience, 1998

[10] H. A. Taha, *Integer Programming*, Academic Press, 1975

[11] H. M. Salkin, *Integer Programming*, Addison-Wesley, 1975

[12] S. Martello, and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, Wiley, 1990

[13] E. Y. Lin, "A bibliographical survey on some well-know non-standard knapsack problems", *INFOR*, vol. 36, no. 4, pp. 274-317, November 1998

[14] D. Pisinger, *Algorithms for Knapsack Problems*, Ph. D. thesis, University of Copenhagen, February 1995

[15] M. R. Garey, and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, 1979

[16] T. Ibaraki, "Enumerative approaches to combinatorial optimization – Part 1", *Annals of Operations Research*, 10

[17] T. Ibaraki, "Enumerative approaches to combinatorial optimization – Part 2", *Annals of Operations Research*, 11

[18] K. Dudzinski, and S. Walukiewicz, "Exact methods for the knapsack problem and its generalizations", *European Journal of Operational Research*, vol. 28, pp. 3-21, 1987

[19] T. E. Gerasch, and P. Y. Wang, "A survey of parallel algorithms for one-dimensional integer knapsack problems", *INFOR*, vol. 32, pp. 163-186, 1994

[20] R. M. Nauss, "The 0-1 knapsack problem with multiple choice constraints", *European Journal of Operational Research*, vol. 2, pp. 125-131, 1978

[21] P. Sinha, and A. A. Zoltners, "The multiple-choice knapsack problem", *Operations Research*, vol. 27, pp. 503-515, 1979

[22] M. E. Dyer, W. O. Riha, and J. Walker, "A hybrid dynamic programming/branch-and-bound algorithm for the multiple-choice knapsack problem", *Journal of Computational and Applied Mathematics*, vol. 58, pp. 43-54, 1995

[23] S. Martello, and P. Toth, "Heuristic algorithm for the multiple knapsack problem", *Computing* 27, pp. 93-112, 1981

[24] A. J. Schwenk, "What is the correct way to seed a knockout tournament?", *The American Mathematical monthly*, vol. 107, no. 2, pp. 140-150, February 2000

[25] B. Ryu, "Modeling and simulation of broadband satellite networks Part II: traffic modeling", *IEEE Comun. Mag.*, pp. 48-56, July 1999

[26] B. A. Barrett, *Traffic Models for Hybrid Satellite-terrestrial Networks*, Master thesis, University of Maryland, 1999, http://www.isr.umd.edu/TechReports/CSHCN/2000/CSHCN_MS_2000-1/CSHCN_MS_2000-1.phtml

[27] H. Koraitim, and S. Tohme, "Resource allocation and connection admission control in satellite networks", *IEEE J. Selec. Areas Commun.*, vol. 17, no. 2, pp. 360-372, February 1999

[28] M. Liu, M. Karir, M. Raissi-Dehkordi, and J. S. Baras, "Hybrid Internet simulation testbed", *ISR Technical Report*, University of Maryland,1999,

http://www.isr.umd.edu/TechReports/CSHCN/1999/CSHCN_TR_99-23/CSHCN_TR_99-23.phtml

[29] S. Rao, "Traffic models for the OPNET simulator of the ALAX", *ISR Technical Report*, University of Maryland, 1995, *http://www.isr.umd.edu/TechReports/CSHCN/1995/CSHCN_TR_95-20/CSHCN_TR_95-20.pdf*

[30] D. Pisinger, "A minimal algorithm for the Multiple-choice Knapsack Problem", *European Journal of Operational Research* 83, pp. 394-410, 1995