Università degli Studi di Roma "La Sapienza"

Dottorato di Ricerca in Ingegneria Informatica

XIV Ciclo – 2001

Scheduling Algorithms and Localization Tools for Wireless Networks

Andrea Vitaletti

Università degli Studi di Roma "La Sapienza"

Dottorato di Ricerca in Ingegneria Informatica

XIV Ciclo - 2001

Andrea Vitaletti

Scheduling Algorithms and Localization Tools for Wireless Networks

Thesis Committee

Prof. Alberto Marchetti Spaccamela (Advisor)
Prof. Stefano Leonardi
Prof. Giancarlo Bongiovanni

Author's address:

Andrea Vitaletti

Dipartimento di Informatica e Sistemistica

Università degli Studi di Roma "La Sapienza"

Via Salaria 113, I-00198 Roma, Italy

e-mail: vitale@dis.uniroma1.it

www: http://www.dis.uniroma1.it/~vitale

Preface

I started my Ph.D. mainly for one reason and with two objectives in mind. The reason was that I like science, particularly in its technological implications, and I love to be free of studying and exploring what I like. *"Personalmente amo investigare liberamente la veritá di quelle affermazioni che mi arrecano piacere"*[1]. This phrase by Sagredo, a good friend of Galileo, effectively expresses what I mean. The objectives were to learn as many things as possible while exploring new solutions, and to show that a stronger cooperation between industry and university is possible. Now, after three years, I can say that the reason was a good one, and that I achieved, in my own small way, both my objectives. This is somehow testified by some publications [26, 59, 54, 75, 69, 51] and three United States provisional patent applications [72, 74, 73].

Acknowledgments

Many friends supported me during these Ph.D. years; I love to say that it is always better to work with friends rather than simply with colleagues.

First I wish to thank Alberto Marchetti-Spaccamela and Stefano Leonardi, I learned a lot from them. I met both of them during the last year in my computer science classes.

I still remember when Alberto during the last lesson of his course in networking said, "... no matter who is the professor and which is the subject of your thesis, it is more important you do something you like ...". Simply the truth.

Stefano is a brilliant computer scientist, and his door has been always open for encouragement and suggestions. More importantly, we had great time eating and drinking [2] together. Wherever you are in the world, he always knows which is the best dish and the appropriate wine.

Thanks also to Luca Becchetti, because the musketeers are always three ... Alexander Dumas docet.

I am grateful to Giorgio Ausiello and Bruno Errico. Both of them, from a different perspective (academic Giorgio and industrial Bruno) gave me the same suggestion when I was dubious in starting my Ph.D., "... It is up to you, but for sure Ph.D. is a great experience ...".

[1] "Personally I love to freely investigate the truth of those assertions that enjoy myself", from "Galileo. A life", James Reston Jr, Harper Collins Publishers, New York, 1994 .

[2] Romans said: *"in vino veritas"*, "In wine the truth".

Thanks to Etnoteam, the company for which I still work, because it gave me enough freedom to do my Ph.D. ... not usual for an Italian company.

Thanks to my AT&T friends Muthu, Rittwik, Ted and Suhas. My stay in the AT&T research labs was really a great experience. In particular thanks to Muthu, for giving me this opportunity and for his infectious joy of living.

Thanks to Camil and Luigi, my roommates. It is not easy to have me and my confusion in the same room.

I thank Stefano, Alberto and Giancarlo Bongiovanni, for serving on my Ph.D. committee.

Thanks to the reviewers Danny Raz, Kirk Prhus and Vincenzo Liberatore, for their suggestions and comments.

I wish to thank also my family and all the other friends for their constant encouragement and moral support.

Finally thanks to Chiara, my wife, because she helps me to be what I am.

Contents

Chapter 1

Introduction

The Internet protocol architecture was originally conceived with the main objective of creating a robust and scalable infrastructure able to support the deployment of a number of applications. The Internet users were assumed to be in a steady state, at the office, at home etc., and accessing the network by means of wired links. Moreover services and applications, mostly required a reliable end-to-end data transfer, satisfactorily supported by a best effort service model. The Internet network described above is no more suitable to face the emerging needs of a new population of Internet nomadic users (see Figure 1.1), such as travellers on wheels, on water or in the air, who desire to gain access to multimedia services regardless of their location and, if possible, while in motion. The deployment of a system capable to support the *ubiquitous Internet*, may benefit from the cooperation of a variety of wireless technologies, such as Wireless Lan, cellular network (GSM/GPRS/UMTS) and satellite networks. In fact neither wireless terrestrial networks nor satellite systems operating by themselves are able to serve the high diversified environments foreseen in the ubiquitous Internet, such as the open rural environment, the suburban/urban environment and the indoor and low-range outdoor environment. The design of a network architecture capable of exploiting and integrating the characteristics of the above wireless technologies is one of the most challenging objective of next generation wireless networks (see Chapter 2).

Today	Ubiquity
	Reachability
	Security
	Convenience
Tomorrow	Localization
	Instant Connectivity
	Personalization

Table 1.1: Attributes of Mobile Communication

1.1 Wireless Internet Applications

A recent study [56] indicates the attributes in Table 1.1 as the key drivers for the increasing sophistication of the mobile market.

In the following we give a brief explanation of these attributes. *Ubiquity:* it is the most obvious advantage of a wireless terminal. A mobile terminal in the form of a smart phone or a communicator can fulfill the need both for real-time information and for communication anywhere, independent of the users location. *Reachability:* with a mobile terminal a user can be contacted anywhere anytime. *Security:* mobile security technology is already emerging in the form of SSL (Secure Socket Layer) technology. Furthermore the smartcard within the terminal, the SIM (Subscriber Identification Module) card, provides authentication of the owner and enables a higher level security than currently is typically achieved in the fixed internet environment. *Convenience:* it is an attribute that characterizes a mobile terminal. Devices store data, are always at hand and are increasingly easy to use. *Instant Connectivity:* instant connectivity to the internet from a mobile phone is becoming a reality already and will fast-forward with the introduction of GPRS services. With WAP or any other microbrowser over GSM, a call to the internet has to be made before applications can be used. Using GPRS it will be easier and faster to access information on the web without booting a PC or connecting a call. *Personalisation:* personalisation is, to a very limited extent, already available today. However, the emerging need for payment mechanisms, combined with availability of personalised information and transaction feeds via mobile portals, will move customisation to new levels, leading ultimately to the mobile device becoming a real life-tool.

Finally *Localization:* the ability to locate the position of a mobile device is considered crucial

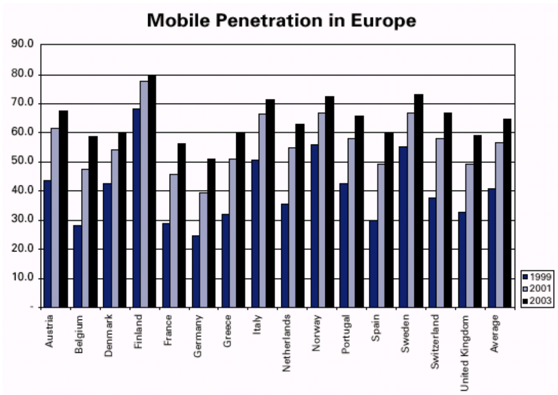


Figure 1.1: Source: Dataquest, Mobile Communications International.

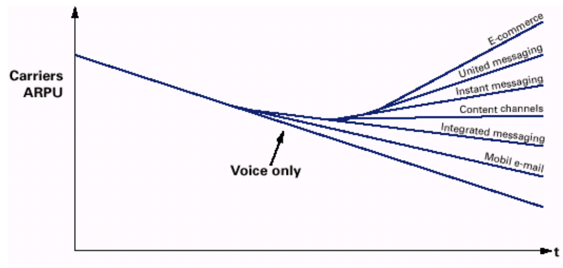


Figure 1.2: Data application are critical. Source: Microsoft.

for providing geographically specific value-added information. Applications using mobile location service technologies include fleet management, vehicle tracking for security, tracking for recovery in event of theft, telemetry, emergency services, location identification, navigation, location based information services and location based advertising. The largest push for localization technology is coming from the US. There, mobile telephone operators have been forced by the FCC to provide emergency 911 services by October 2001 in such a way that the location of the caller could be determined within a radius of 125 meters in 67% of all cases. There are three major localization techniques: 1) *Triangulation* can be done via *lateration*, which uses multiple distance measurements between known points, or via *angulation*, which measures angle or bearing relative to points with known separation; 2) *Proximity* measures nearness to a known set of points; 3) *Scene analysis* examines a view from a particular vantage point, such as antennae or mobile terminals. Implementation of location systems generally uses one or more of these techniques to locate objects and people. Furthermore location systems can be either *terminal based* or *network based*. In terminal based systems, it is the mobile device itself that determines the location. Normally these systems are less accurate but they do not require significant network upgrade. Network based systems provide more accurate localization information, but require a significant upgrade of the network. In chapter 6 of this thesis we present our terminal based localization technique. This technique is protected by three United States provisional patent applications ([72], [74], [73]).

1.2 Quality of Services for new wireless applications

With mobile communications reaching the mass market, network operators are facing decreasing ARPU (Average Revenue Per User, See Figure 1.2). The network operators must then continuously implement new services on their network. These new services (mainly data services) often require a suitable *Quality of Services* (QoS). Nowadays users are not simply satisfied by the availability of the service itself, they want to experience a good QoS. A real time application, such as for example video conference, not only imposes strict requirements in terms of bandwidth, but also in terms of packet delay (normally should be less than 400ms) and jitter (jitter is the variation of delay and should be almost constant). On the contrary, a File Transfer application, has mainly requirements in terms of bandwidth, being extremely tolerant to high delay and variable jitter. The heterogeneous nature of the Internet applications, requires an effort in developing a network infrastructure capable of addressing such requirements.

The best effort QoS currently available over the Internet, is unable to effectively cope with these requirements. From a best effort point of view, each IP packet is handled in the same way, independently if it has been generated from a real time application, rather than from a simple FTP. All the IP packets scheduled by a router, are queued in the same queue and are typically served according to a FIFO policy; thus Real time traffic may be delayed because of the presence of a huge number of FTP packets. In a best effort network, the only way to guarantee the QoS is over-dimensioning network links, in order to avoid traffic congestion.

A first solution proposed to address the above problems is the introduction of the DiffServ architecture. In DiffServ, each packet is classified according to a specific classification policy, for example distinguishing among real time packets and FTP packets. Each packet in a specific class is handled according to a particular scheduling algorithm. For example, real time packets can have a priority over FTP packets. In such a way the delay accumulated by real time packets is not anymore influenced by the presence of FTP packets. Note however that this architecture does not avoid the generation of delay within the real time class itself. In other words, if the number of real time sessions become too high, it is still possible the accumulation of a huge delay. Although DiffServ is a simple and scalable solution offering a “better than best effort” QoS, it potentially suffers of the same problem of congestion of best effort. A further solution capable of addressing this problem, is the IntServ architecture. The IntServ architecture, expects that each new communication flow reserves in advance (for example through the RSVP

protocol) the resources for providing a suitable QoS. In other words, the system checks if it has enough resources to manage the new communication and to simultaneously maintain a suitable QoS for the ongoing communications. If those resources are available, the new communication is accepted, on the contrary it is refused or it is accepted with a reduced QoS compatible with the available resources. Although IntServ effectively provide an "hard QoS" to the accepted communications, because of its per-flow orientation is extremely resource consuming and can be applied only within small scale networks. For this reason, it has been proposed an hybrid approach taking advantage from the cooperation of IntServ and DiffServ (see Chapter 2).
In this thesis we mainly focus on network layer QoS, nevertheless similar issues arise at link layer. Cooperation between network and link layers to achive overall QoS is in part still an open problem.

1.3 Wireless Scheduling

A relevant part of the above cited QoS mechanisms is taken by traffic scheduling policies, at which is dedicated a considerable part of this thesis (see Chapters 3, 4 and 5). A scheduling algorithm allocates resources to communication requests in order to minimize/maximize an objective function that normally describes the QoS experienced by the users. As we have seen above, it is a scheduling algorithm that, according to a specific scheduling policy (from the simple FIFO in best effort to the more complicated priority queuing in DiffServ enabled routers), schedules the next packet to be served in a router.
In the following we introduce the main characteristics of scheduling algorithms focusing on the peculiarity of the wireless environment. We have seen that the demand for wireless communication systems is continuously growing (see Figure 1.1). Since wireless frequency is a scarce resource, efficient frequency utilization is a relevant issue; although new wireless technology greatly increase the bandwidth availability of wireless systems, this is still a limited fraction of the bandwidth available in wireline networks. Resource allocation schemes and scheduling policies are critical to optimize frequency utilization. However, resource allocation and scheduling schemes from the wireline domain cannot be directly applied to wireless system, since *wireless channels have unique characteristics*, such as limited bandwidth, time-varying and location-dependent channel condition and channel-condition-dependent throughput. In particular, if we

consider the radio propagation, it can be roughly characterized by three nearly independent phenomena: path-loss variation with distance (path losses vary with the movement of mobile stations), slow log-normal shadowing and fast multipath-fading. The last two characteristic are time-varying and they imply that users perceive time-varying service quality. For voice users, better channel conditions may results in better voice quality. For data users, better channel condition can be used to provide higher data rates. A good scheduling scheme should be able to exploit the time-varying channel conditions of users to achieve higher utilization of wireless resources. Nevertheless the potential to exploit higher data throughput introduces the trade-off problem between wireless resource efficiency and fairness among users. Since wireless spectrum is a scarce resource, improving the efficiency of spectrum utilization is a crucial issue. However a scheme designed only to maximize the overall throughput could be unfair among users. For example allowing only users with good channel condition to transmit with high transmission power may result in a very high throughput, but it is unfair to other users. In this thesis we will mainly consider objective functions that express user satisfaction. For example we will consider minimizing the maximum or average response time (the response time is the time elapsed between the release of a request and its completion). Other possible objective functions are minimizing the total weighted response time, or minimizing maximum or total stretch (whereas the stretch of a job is the ratio between its flow time and its processing time or size).

The scheduling problem that we consider are NP-hard optimization problems and, therefore, we do not expect to solve them exactly in polynomial time. For this reason we are interested in algorithms that provide us an approximate solution in polynomial time. We will measure the quality of the approximate solution using both an experimental approach and/or evaluating theoretically the performance obtained by the algorithms. Scheduling algorithms can be considered in two main variants, namely *offline* and *online*. The offline version of a scheduling problem assume that all requests are known in advance before being served. Hence the offline algorithm has a full knowledge of the future requests and can schedule them in the most appropriate way in order to minimize/maximize the objective function. The offline case is of theoretical interest and is mainly useful to quantify the benefit to be accrued from scheduling, that in most practical cases must be necessarily online.

In the online version of a scheduling problem, requests arrive over time, and scheduling algorithms have to take their decisions without knowledge of future requests. A standard technique used to evaluate the performances of the online algorithms is the *competitive analysis*, whereby the quality of an online algorithm on each input sequence is measured by comparing its performance to that of an optimal offline algorithm. In our work we will also use *resource augmentation analysis*, a different technique to evaluate the performance of an online algorithm. Resource augmentation allows the online algorithm to schedule the input requests, possibly less resource demanding with respect to the original input, having more system resources available. Why should one compare the performance of an algorithm to an adversary (the offline algorithm), if the algorithm is given more resources than the adversary? First, this technique allows us to analyze algorithms that would be hard to analyze using competitive analysis. Secondly resource augmentation based analysis gives an indication of the amount of extra resources needed in order to obtain a certain, guaranteed QoS. This information can be used during system design to better dimension network links and devices. Finally, resource augmentation provides a tool to design new algorithms that perform well in practice, whereas worst case analysis would suggest that they have a poor behaviour in theory.

1.4 Overview of the Thesis

This thesis is structured in three main parts: 1) Suited: A Prospect of QoS Enabled Wireless Communication and Services (Chapter 2), 2) Scheduling problems (Chapters 3, 4 and 5), 3) system architecture for location based services (Chapter 6).

1.4.1 Suited: A Prospect of QoS Enabled Wireless Communication and Services

Some of the problems that we study in this thesis have been inspired by the participation of the author to the EU research project Suited (multi-segment System for broadband Ubiquitous access to InTErnet services and Demonstrator). The main goal of Suited is the design and the deployment of an architecture able to support multimedia application with a guaranteed QoS, irrespective of the location of mobile users. Suited represents a valuable opportunity to better understand the main problems in the emerging areas of wireless communications. Furthermore the participation to Suited gave the author the great opportunity to place his study in a realistic context; some of the problems that we present in this thesis, even those ones that received a more theoretical handle, have been inspired by this participation. This section describes the objectives of Suited, the main problems that we encountered in the design of the Suited Architecture and outlines the solutions proposed to address those problems. Hence this chapter can be considered as an overview of the main problems and possibly architectural solutions of a tomorrow QoS enabled wireless/wired-integrated Internet.

1.4.2 Scheduling problems

The second part of the thesis is devoted to the study of some interesting scheduling problems. In the following we briefly describe these scheduling problems and we discuss the main achieved results.

Downlink Scheduling for Multirate Wireless Networks

There is tremendous momentum in the wireless industry towards next generation (3G and beyond) systems. These systems will not only migrate the existing voice traffic to a higher bandwidth platform, but are also expected to jumpstart large scale data traffic. Our focus is on the downlink channel performance, which is likely to be a major focus in emerging systems since data traffic is expected to dominate over time and data traffic typically tends to have asymmetrically large downlink demand. Next generation 3G/4G wireless data networks, such as CDMA, allow multiple codes (or channels) to be allocated to a single user, where each code can support multiple data rates (data rates are function of the channel condition and transmission power). This results in more flexibility than is available in current systems to manage and modulate the traffic. Furthermore this gives rise to a new class of scheduling problems. Our main contributions are threefold.

- We abstract a general downlink scheduling problem which has many novelties. For example, we embody channel characteristics guided by communication theoretic considerations, and the properties of these channels get exploited in our scheduling algorithms. We study QoS parameters related to per request behavior, in particular, we focus on optimizing response time per request. In contrast, prior work in wireless systems scheduling has typically focused on rate optimization metrics.
- The scheduling problems that arise above are hard to solve exactly since we show them to be NP-complete. However, we use an unusual analysis technique: *resource augmented competitive analysis*, to derive simple, online algorithms which are not only practical, but

also have provably good performance in approximating the optimal maximum response time of a job.

- We present a detailed experimental study of our algorithms. Using real web server request logs and realistic 3G/4G system parameters, we show experimentally that our online algorithms perform significantly better than our worst-case analyses indicate. Moreover our results indicate, the proposed scheduling algorithms can pack power and codes effectively, that is, they benefit from the multiple code, multi-rate feature of 3G/4G systems.

This work combines aspects of combinatorial optimization (convex programming), scheduling algorithms (analyzing online algorithms with augmented resources), and applies them to general scheduling problems that arise in next generation wireless systems.

Bandwidth and Storage Allocation Problems under Real Time Constraints

The problem we study has been encountered in the context of the EU research project Euromednet on scheduling requests for remote medical consulting on a shared satellite UDP-TCP/IP channel. Every request asks for a number of contiguous bandwidth slots to provide every request with a UDP-TCP/IP satellite connection between the users involved in the consulting. Bandwidth is assigned in slots of 64 kb/sec. The number of slots per end user depends on the type of service desired (typical values are 64 kb/sec for common internet services – 384 Kb/sec for audio/video.) At most 48 slots of 64 Kb/sec are available on the channel in this specific application. Requests also specify a duration of the consulting (typical values are from 1/2 hour to 2 hours), to be allocated within a time interval specified in the request. Requests, that are typically issued a few days in advance, are replied soon by the system with a positive or a negative answer on the basis of the pending requests and of the resources already allocated. Every accepted request is allocated starting from a base bandwidth for a contiguous number of slots along a time duration within the indicated time interval. The total bandwidth assigned to a single request must be contiguous due to the constraints imposed from FDMA (Frequency Division Multiple Access) technology. Our main contributions are as follows.

- We abstract an interesting combinatorial problem from the problem encountered in this application : every accepted request is scheduled on a rectangle in the time/bandwidth Cartesian space of basis equal to the duration and height equal to the requested bandwidth.

Accepted requests must observe the *packing constraint* imposing no overlaps between any two scheduled rectangles. A benefit associated with every request indicates its relevance or the economic revenue gained from its acceptance. The objective is to maximize the overall benefit obtained from accepted requests.

- We present a constant approximation algorithm for the RP problem; this is currently the best approximation for the RP problem. This result is obtained by solving a fractional LP relaxation and applying a novel rounding technique to the optimal solution of the LP. We also show how to replace the solution of the fractional LP relaxation with a combinatorial algorithm.

Randomized Lower Bounds for Online Path Coloring

This is the most theoretical problem we study in this thesis. We provide lower bounds for online interval graph coloring and for online path coloring on tree networks. This abstracts a set of scheduling problems in wireless communications. Requests arrive over time. Each request specifies a contiguous time interval in which it has to be served. According to the FDMA technology, two overlapping requests must use different frequencies, namely two overlapping intervals must be colored with different colors. The problem is on-line since the assignment of frequencies to requests must be done upon arrival even if the requests have to be served in the future. The goal of the algorithm is to schedule (color) all the input requests using as few frequencies (colors) as possible. This problem is equivalent to the well known problem of coloring the vertices of an interval graph. In particular, in this thesis we study the power of randomization in the design of online path coloring algorithms. Our main contributions are the following.

- We show that no randomized algorithm for online coloring of interval graphs achieves a competitive ratio strictly better than the best known deterministic algorithm.

- We also present a first lower bound on the competitive ratio of randomized algorithms for path coloring on tree networks. We prove an $\Omega(log\Delta)$ lower bound for trees of diameter $\Delta = O(logn)$ that compares with the known $O(logn)$-competitive deterministic algorithm for the problem.

1.4.3 System architecture for location based services

CDPD has been deployed in North America for providing data services for mobile users. We present a system architecture whose purpose is extracting location information of a CDPD subscriber from the handheld device and making it available to a location server. The location server can then be accessed by an Internet Service Provider (ISP) in order to offer suitable location based services. The basic localization technique we implemented, i.e. the BSIC localization method, allows us to identify the location of a user within the range of a cell. The main idea of this method is that through a suitable protocol, the MSCI protocol, we can obtain from the modem the Base Station Identification Code (BSIC), namely a number that unambiguously identifies the antenna to which the user is currently connected. Since an antenna covers a specific region (i.e. a cell), the BSIC also identifies the cell in which the mobile user is currently located. The information on location can be packaged in a way to allow the subscribers to download a web page from a portal customized according to their locations. For example, location information can be exploited by Yahoo yellow pages in order to offer the closest emergency services or attractions to the mobile users. We have designed and tested a System Architecture cable to provide location aware service exploiting the localization information obtained by the BSIC method, and our main contributions are as follows.

- We implemented a simple handset assisted method, i.e. the BSIC method, to localize a mobile user. This approach, in contrast to the network oriented localization solutions, has a minimal impact on the telecommunication network and protects privacy (the user has to explicitly disclose his localization information).

- The BSIC localization technique suffers from an inherent lack of *accuracy*, since the dimensions of a cell vary between some meters and some kilometers. We compare and contrast the accuracy figures of this technique with those of a global positioning system (GPS) in order to determine the applicability of the different localization techniques. For example, in a metropolitan area the use of a CDPD based localization technique can be sufficient to the purposes of general directory services. However, if a user wants to be routed to a specific point of interest a GPS is needed since the accuracy derived from a CDPD system is not sufficient. The comparison has been done running some experiments in three contexts: a metropolitan area, a suburban area and finally an highway.

- Finally we implemented and tested two further localization techniques, the multiple cells method and the RSSI method, designed to improve the accuracy of the BSIC method. Also in this case, the accuracy of these techniques has been tested with respect to the GPS accuracy running some experiments in different contexts.

Chapter 2

Suited: A Prospect of QoS Enabled Wireless Communication and Services

The Internet protocol architecture was originally conceived with the main objective of creating a robust and scalable infrastructure able to support the deployment of a number of applications. The Internet users were assumed to be in a fixed location, at the office, at home etc., and accessing the network by means of wired links. Moreover, services and applications mostly required a reliable end-to-end data transfer, satisfactorily supported by a best effort service model. The Internet network described above is no more suitable to face the emerging needs of a new population of Internet nomadic users, such as travelers on wheels, on water or in the air, who desire to gain access to multimedia services regardless of their location and, if possible, while in motion. Moreover, the best effort quality of the services currently available over the Internet network is nowadays becoming unsatisfactory for several users' categories.

The growing needs of these new classes of Internet users requiring to access multimedia applications irrespective of the location and with the desired Quality of Service (QoS), are addressed in the *Suited (multi-segment System for broadband Ubiquitous access to InTErnet services and Demonstrator)* project [41].

The participation to Suited gave the author the opportunity to better understand the main problems in the emerging area of wireless communication and to place his study in a realistic context. Some of the problems that we present in this thesis, even those ones that received a more theoretical handle, have been inspired by this participation.

This chapter describes the objectives of Suited, the main problems that we encountered in the design of the Suited Architecture and outlines the solutions proposed to address those problems. Observe that a wireless device [1], interacts with the wired infrastructure that physically provides the connection between wireless mobile users. Furthermore a major goal of Suited is the provisioning of QoS enabled services. Hence this chapter can be considered as an overview of the main problems and possibly architectural solutions of a tomorrow QoS enabled wireless/wired-integrated Internet.

2.1 Suited and the GMBS

Suited aims at contributing to the design and deployment of the Global Mobile Broadband System (GMBS) [26] , the essential ingredient of the ubiquitous Internet. The service area of the GMBS system shall be world-wide and available in high diversified environments such as: the open rural environment, the suburban/urban environment and finally the indoor and low-range outdoor environment. Nevertheless, neither wireless terrestrial networks nor satellite systems operating by themselves are able to serve such a wide range of areas. In order to overcome this issue, the solution proposed in Suited foresee that a *multi-segment access network*, whose components present mutually complementary features (see table 2.1), shall inter-work with the *Federated Internet Service Provider (ISP) network*. The Federated ISP network, consists of a set of ISPs which have agreed peer Service Level Agreement (SLAs). A SLA defines the basic characteristics in terms of connectivity the user needs from the network to provide mobile Qos sensitive Internet services. From a user perspective, the GMBS system is perceived as a single network able to support mobile and portable, Qos guaranteed, Internet services.

2.1.1 Wireless Technologies

The multi-segment access network may benefit from the cooperation of a variety of wireless technologies. In the following we give a brief description of the main current wireless technologies.

[1] except in ad-hoc networks.

Environment	Low data rate services (up to 64Kbps)	Medium data rate services (up to 150Kbps)	High data rate services (up to 512Kbps)
Indoor/Low range outdoor	WLAN, GPRS/UMTS	WLAN, GPRS/UMTS	WLAN, GPRS/UMTS
Outdoor Urban/suburban	SAT, GPRS/UMTS	SAT, GPRS/UMTS	SAT, GPRS/UMTS
Outdoor Open area/rural	SAT, GPRS/UMTS	SAT, GPRS/UMTS	SAT, GPRS/UMTS

Table 2.1: Segment Availability per Service Rate in Different Service Environments (Suited context).

- *Ka-band satellites* Ka-band satellites operates in the range of 18 to 31 GHz. They are mainly used by multimedia companies because they offer sufficient bandwidth (up to 2Mbps for each link) to support multimedia applications. The wide range of frequencies allows data to be transmitted at multiple frequencies simultaneously and allows for two-way broadband services . In this summary, we focus on two satellites technologies: Geostationary satellites (GEO) and Low-Earth Orbit (LEO) satellites. GEO satellites orbit 36000 Km from the Earth, which means they orbit at the same speed of the Earth's rotation, keeping them above the same spot on Earth. GEO satellites have the best advantage for reaching the largest amount of the Earth's surface. The long response time is the greatest disadvantage of these systems; the round-trip propagation delay for a GEO transmission is about 260 ms while a LEO (Low-Earth Orbit) transmission only has a 10 ms delay (LEO's latency is comparable to that of wide area terrestrial links). LEO satellites orbit closest to Earth's atmosphere (700-1400 Km). Being the closest satellites to Earth they can only reach a relatively small area; this means providing world-wide service requires more satellites than those in GEO orbit. Furthermore the relative position of a LEO satellite with respect to a ground user is constantly changing. In addition, satellites in this orbit cost less to get started, but they only last 5-8 years. The small round trip of LEO satellites is very attractive when providing service for real time applications since the users will notice the delay from a GEO link, but may not even be aware of a delay on a LEO link. The answer to propagation delay for GEO satellite data networks is the use of advanced protocols, such as High Level Data Link Control (HDLC) and Synchronous Data Link Control (SDLC), and/or delay compensators. They provide acknowledgments locally before data is transmitted over the satellite, thus eliminating the lag time for protocol handshakes.

- *GSM* GSM (Global System for Mobile Communication) operates in the 900 MHz and the 1800 MHz (1900 MHz in the US) frequency band and is the prevailing mobile standard in Europe and most of the Asia-Pacific region. GSM is used by more than 215 million people (October 1999), i.e. representing more than 50% of the world's mobile phone subscribers. North America has only about 5 million GSM users in late 1999, while the majority of subscribers are using a variety of technologies for mobile communications, including pagers and a high percentage of analogue devices.

- *HSCSD* HSCSD (High Speed Circuit Switched Data) is a circuit switched protocol based on GSM. It is able to transmit data up to 4 times the speed of the typical theoretical wireless transmission rate of 14.4 Kbit/s, i.e. 57.6 Kbit/s, simply by using 4 radio channels simultaneously. In total there are only 18 GSM operators worldwide who intend to offer HSCSD service, before they introduce GPRS (the definition is forthcoming in the next section). The key problem in the emergence of this market is that there is currently only Nokia who can provide PCMCIA modem cards (CardPhone 2.0) for HSCSD clients, which offers a transmission speed of 42.3 Kbit/s downstream and 28.8 Kbit/s upstream. The typical terminal for HSCSD is a mobile PC rather than a smartphone. Call set-up time is still 40 seconds needed for the handshake of the modem.

- *GPRS* GPRS (General Packet Radio Service) is a packet switched wireless protocol as defined in the GSM standard that offers instant access to data networks. It will permit burst transmission speeds of up to 115 Kbit/s (or theoretically even 171 Kbit/s) when it is completely rolled out. The real advantage of GPRS is that it provides an always on connection (i.e. instant IP connectivity) between the mobile terminal and the network. Network capacity is only used when data is actually transmitted. The actual speed of GPRS will be initially a lot less than the above dream figures: 43.2 Kbit/s downstream and 14.4 Kbit/s upstream up to 56 Kbit/s bi-directional some time thereafter.

- *EDGE* Enhanced Data Rates for Global Evolution (EDGE) is a higher bandwidth version of GPRS permitting transmission speeds of up to 384 Kbit/s. It is also an evolution of the old GSM standard and will be available in the market for deployment by existing GSM operators during 2002. Deploying EDGE will allow mobile network operators to offer high-speed, mobile multimedia applications. Furthermore EDGE allows a migration path from GPRS to UMTS.

- *3G* 3rd generation (3G) is the generic term for the next big step in mobile technology development. The formal standard for 3G is the IMT-2000 (International Mobile Telecommunications 2000). This standard has been pushed by the different developer communities: W-CDMA as backed by Ericsson, Nokia and Japanese handset manufacturers and cdma2000 as backed by the US vendors Qualcomm and Lucent. The goal of being able to have one single network standard (CDMA) and use one handset throughout the world seems to be capable of being reached. But within the one standard there will be 3 optional, harmonized modes (W-CDMA for Europe and the Asian GSM countries, Multicarrier CDMA for North America and TDD/CDMA for the Chinese). UMTS (Universal Mobile Telephone System) is the third generation mobile phone system that will be commercially available from 2003 in Europe. Although many people associate UMTS with a speed of 2 Mbit/s, this will be reached only within a networked building and indeed only with some further development to the technology. Realistic expectations suggest a maximum capacity in metropolitan areas of 384 Kbit/s , at least until 2005. This is in fact the same transmission rate that can be realized much earlier with EDGE.

- *Wireless LAN (WLAN) segment* We just consider the IEEE 802.11 standard. IEEE 802.11 makes provisions for data rates up to 11 Mbps (802.11b), and calls for operation in the 2.4 - 2.4835 GHz frequency band (in the case of spread-spectrum transmission), which is an unlicensed band for industrial, scientific, and medical (ISM) applications. There are two different ways to configure a wireless network: ad-hoc and infrastructure. In the ad-hoc network, computers are brought together to form a network "on the fly".; there is no structure to the network, there are no fixed points, and usually every node is able to communicate with every other node. A good example of this is a meeting where employees bring laptop computers together to communicate and share design or financial information. The infrastructure architecture uses fixed network access points through which mobile nodes can communicate. These network access points, coordinate the communications, and normally are linked to the corporate LAN connecting wireless nodes to other wired nodes. A wireless LAN normally provides a short range connectivity mainly used for indoor connections.

As far as Suited multi-segment access network concerns, it benefits from the cooperation (see table 2.1)of WLAN, GPRS/UMTS and the EuroSkyWay M-ESW (i.e., a constellation of Ka LEO and GEO satellites).

2.1.2 Internet Service Requirements

Suited is intended to support a wide set of ISP services; in the following we just mention the main of them:

- best effort: the traditional Internet services such as Web browsing, e-mail, telnet, ftp.
- playback: audio and video streaming over IP (radio webcasting, TV webcasting, etc.)
- real-time: audio and video real time over IP (i.e. video, audio, data conferencing based on the T.120, H.323 and SIP standards)
- support services: Directory Services (to share resources and to improve searching features), location based services etc.

All these services, to satisfy the new demands of the Internet users, must be deployed in a QoS enabled system, namely a system capable to guarantee a suitable QoS. While there has been significant progress in the design of Quality of Services (QoS) architecture to improve the present best effort Internet, there are a still number of QoS aspects that appear to need further investigation. In particular it remains to define a suitable, common adopted framework for the definition and evaluation of the QoS Internet requirements. Nowadays Internet applications are extremely heterogeneous in terms of QoS requirements. For example if we limit our attention to bandwidth requirements (see Figure 2.1), we can observe: 1) some applications show an asymmetry between the uplink and downlink bandwidth requirements, 2) the bandwidth required by the applications may vary from some Kbps to some Mbps.

A new approach for evaluating the Internet Service Requirements. In Suited we proposed a quite innovative approach [59] for evaluating the Internet service requirements. This approach is derived from the 3x3 matrix evaluation approach described in the ITU-T [42], with the addition of the fourth column, i.e. the *security*. QoS parameters, and their relevant requirements, are obtained considering the four performance criteria (Speed, Accuracy, Dependability, Security) applied to each communication functions (Access User Information Transfer,

Figure 2.1: Multimedia Bandwidth Requirements.

Disengagement). In Table 2.2 we summarized the main QoS concepts involved in the ITU-T performance framework. For example, the access function requirements concerning a web browsing session can be evaluated in terms of requirements relevant to access speed, access accuracy, access dependability and access security while accessing a nominal web page. There are not specific suggestions to determine the figures for the ITU-T performance criteria in the IP context (i.e. speed, accuracy and dependability), hence, in [59] we suggest to determine those values according to the following criteria:

- C1) Current QoS performances, achieved on best effort IP networks through a 56Kbps dial-up connection, must represent a lower bound to GMBS QoS requirements;
- C2) Application QoS requirements; e.g. to support a audio/video conference with a 6 inches picture size, a frame rate of 30 fps (frame per second) and audio toll quality, we need at least 384Kbps;
- C3) Wireless technologies limits impose an upper bound on QoS performances, e.g. a GPRS link can support at most about 150Kbps, while a satellite link can support up to 2Mbps;
- C4) User expected performances are subjective wishes that should be taken into account

Communication functions	
Access	The access function begins upon issuance of an access request signal or its implied equivalent at the interface between a user and the communication network. It ends when either:1) a ready for data or equivalent signal is issued to the calling users; or2) at least one bit of user information is input to the network (after connection establishment in connection-oriented services). It includes all activities traditionally associated with physical circuit establishment (e.g. dialing, switching, and ringing) as well as any activities performed at higher protocol layers.
User information transfer	The user information transfer begins on completion of the access function, and ends when the “disengagement request” terminating a communication session is issued. It includes all formatting, transmission, storage, error control and media conversion operations performed on the user information during this period, including necessary retransmission within the network.
Disengagement	There is a disengagement function associated with each participant in a communication session: each disengagement function begins on issuance of a disengagement request signal. The disengagement function ends, for each user, when the network resources dedicated to that user's participation in the communication session have been released. Disengagement includes both physical circuit disconnection (when required) and higher-level protocol termination activities.
Performance criteria	
Speed	Speed is the performance criterion that describes the time interval that is used to perform the function or the rate at which the function is performed. (The function may or may not be performed with the desired accuracy.)
Accuracy	Accuracy is the performance criterion that describes the degree of correctness with which the function is performed. (The function may or may not be performed with the desired speed.)
Dependability	Dependability is the performance criterion that describes the degree of certainty (or surety) with which the function is performed regardless of speed or accuracy, but within a given observation interval.
Security	Security is the performance criterion that describes the degree of confidentiality (or secrecy) with which the function is performed regardless of speed or accuracy and dependability.

Table 2.2: QoS and Performance concepts in ITU-T Framework (I.350 Serie)

in the definition of QoS requirements.

To clarify the above concepts, lets consider a video streaming session. Suppose we would offer a video streaming with a 6 inches picture size, 30fps and audio toll quality, this implies that we need at least a 384Kbps link (C2). This means that in this case, the Speed of the User information transfer function would be 384Kbps. For some users, a video streaming with a 6 inches picture size, 15fps and audio toll quality may be adequate. In this case we reduce the bandwidth requirements to 128Kbps (C4). In any case, a video streaming on GPRS cannot support connection speed greater than 150Kbps, while satellite can support up to 2Mbps speed connection (C3). Let consider another scenario, i.e. an ftp session to download a 3MB mp3 file. If we consider a 56Kbps wired connection, it allows to download the file in about 6 minutes (C1). If on the contrary, we consider a 2Mbps satellite connection, the file can be downloaded in only 12 seconds (C3). For a common user may be acceptable to download the mp3 file in 3 minutes, which means a download speed of 128Kbps (C4).

Service Level Agreement Now that we have defined a framework to evaluate the service requirements, we need a contract to enforce those requirements. In the Internet, this contract is the Service Level Agreement (SLA). A SLA specifies how a given sub-network must handle the traffic that receives from an upstream one. The typical SLA that is currently stipulated between ISPs (Internet Service Providers) and end-users specifies only two QoS parameters, i.e. bandwidth and availability.

A Bandwidth Guarantee SLA guarantees the Client that the ISP network will not be a bottleneck for communications with the Client. In particular, in a typical contract the ISP guarantees a minimum bandwidth of 95% of the nominal speed of the access port. An Availability Guarantee SLA guarantees the Client that the service will be available for a minimum of 99.9% of the contractual period.

This kind of SLA is the only possible in the current Internet where best-effort is the dominant service model. In fact best effort does not provide any mechanism to guarantee QoS. The only way to respect the SLA is over dimensioning the network links. Moreover this kind of SLAs is often inadequate to support a suitable QoS.

In Suited we outline a possible SLA migration path that starting from the current “SLA situation” evolves towards a more structured and powerful Internet Service application platform. We first observe that the basic SLA described above does not provide any requirement in terms of delay and packet loss. The evolution of the SLA must consider these requirements, in particular packet delay that represents one of the most important factor that influences the performances experienced by the users (especially in real time applications) . Observe that delay is strictly related to packet loss. In several applications, like real time applications, in which a delay guarantee is the critical part of QoS experienced by the user, packets that do not obey the delay requirements are discarded. A further extension of the SLA is the introduction of jitter guarantee. Jitter is the variation of delay over time and highly affects the quality of real time application. A constant jitter is very important to guarantee a good quality of audio and video real time application.

Up to now we have considered SLAs that do not distinguish between different kinds of traffic, namely Web traffic, e-mail traffic, ftp traffic or real time traffic is handled at the same way. Requirements in terms of bandwidth, availability, delay, packet loss or jitter, are applied without any distinction to all the IP packets. This makes the contract less flexible and also may lead to overcharging traffic with low QoS requirements. For instance packet delay and packet loss are of great relevance for real time applications while are much less relevant for data transfer. On the other hand applications like e-mail and telnet are much less bandwidth demanding than video and voice. A possible extension of the SLA dealing with the above considerations consists in the agreement of separate SLAs for different type of traffic. The user specifies by a traffic descriptor (TD) the traffic that he will generate. [2] If the user injects into the network a traffic compliant with the TD, then the network is capable to guarantee the QoS parameters in terms of bandwidth, delay, packet loss and jitter. On the contrary, if the user injects into the network non-compliant traffic, the network limits the accepted traffic in accordance with the specified TD (for example discarding packets belonging to misbehaving flows or assigning those packets to a lower service level).

A final extension of the SLA is the support of user mobility while guaranteeing a suitable QoS. The mobility SLA may be specialized in two main basic mobility scenarios: Inter-segment handover and Intra-segment handover. The inter-segment handover is between two link of the same wireless segment. A classical example is handover between two wireless LAN access points within a building or a campus, or handover between two different cellular cells. Intra-segment handover is handover between different wireless segment . For instance the handover from a cellular link to a satellite link. We foresee two main level of QoS handover: QoS-aware handover and smooth handover. The Qos-aware handover is performed without degradation of the QoS. In smooth handover, users accept a momentary degradation of performances during the handover, but the session remains active and the QoS levels are quickly restored.

2.1.3 Federated ISP and the nowadays Internet

Best effort is the dominant paradigm in the current Internet. Although most of the commercial routers support sophisticated service models capable to improve the QoS through suitable mechanisms (i.e. DiffServ, Intserv), nevertheless only few ISPs exploit these functionality in a commercial environment. As far as mobility concerns, it is becoming a need with the 3G wireless network spreading, nevertheless most of the current ISPs do not implement any specific mechanism to facilitate mobile users. In fact, normally mobile users access Internet by means

[2] Dual Leaky Bucket parameters represent a standard way to describe the network traffic in terms of Peak Cell Rate (PCR) and its tolerance, and Sustainable Cell Rate (SCR, i.e. the average rate) and its tolerance.

of the access gateway of a single national provider (typically the GSM/GPRS provider), thus they do not need any specific mobile IP features. Indeed users gain an IP address belonging to the GSM/GPRS provider domain, and this address and the access point to Internet, remain always the same during the session, regardless of user location. This restriction will become unacceptable in a fully mobile environment, in which users will be free to move over countries. RFC2002 [65] has been released to support mobility in IPv4, while Perkins and Johnson [66] proposed an IPv6 based solution, however there is not yet a commonly adopted standard. The above considerations show that today Internet is unable to meet the emerging needs in terms of mobility and QoS. This justify the development of a new environment designed to be a first step toward the QoS aware mobile Internet. This is the main goal of the Federated ISP, a federation of Service Provider with the common purpose to provide mobility and QoS in a commercial environment.

2.2 Suited System Architecture

Suited must allow mobile Internet users to access multimedia applications irrespective of their location. Moreover mobile users expect to receive from the Federated ISP QoS assurances; at least the same performances offered by current wired Internet must be guaranteed. To reach these objectives, the main features required to the Suited network architecture can be summarized in three points: 1) Best effort support, 2) Qos support, 3) Mobility support. While best effort support is a standard features of the Internet architecture and it does not require the developing of new solutions, QoS and Mobility support require a significant research and development activity to adapt existing solutions to our framework, along with innovative ways of operation. The Federated ISP is made of two main portions (see Figure 2.2):

- *Edge Network.* It is the integration of the multi-segment wireless access network and the Edge terrestrial subnetworks. It allows mobile user to access the Federated ISP in a IntServ environment.

- *Core Network.* It provides connectivity between the Edge subnetworks in a DiffServ environment. It also provides connectivity to the Internet.

This architecture is based on a hybrid IntServ-Diffserv approach; IntServ is designed to reserve resources creating a virtual wire and thus it offers quantifiable performances, Diffserv does not assure "hard QoS", but it guarantees scalability. In the following we describe in more detail the main features of the Edge and Core Networks.

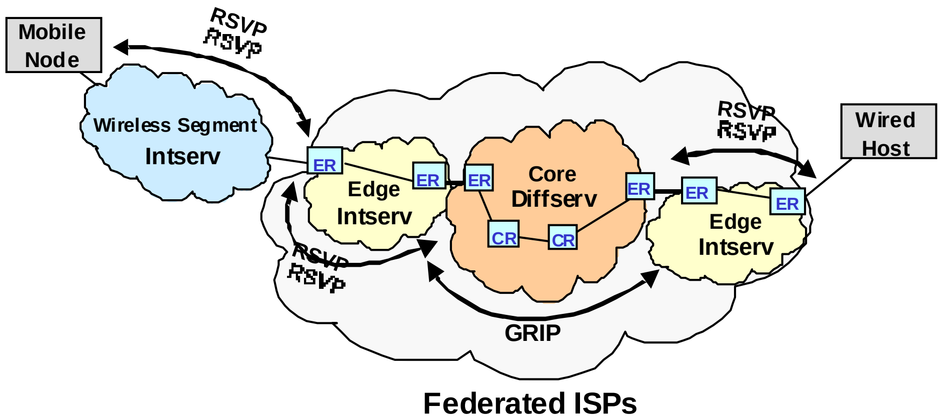


Figure 2.2: Federated ISP Network Architecture.

2.2.1 Edge Network

The edge network is the point at which local mobile-users are allowed into the Federated ISP. In the edge network we adopted the IntServ service model , that allow us to provide "hard QoS guarantees", namely it can assure suitable levels of QoS to the applications . In IntServ, sender-receiver pairs reserve (by means of the signalling protocol RSVP [20, 93, 89]) some resources in each network element in order to guarantee an upper bounded end-to-end queuing delay and no packet loss for each new application flow. The main advantages of the IntServ approach are the following: 1) it provides "hard QoS guarantees", 2) it is supported in many commercial Real-time application (e.g. Microsoft NetMeeting) 3) it is implemented in most of the commercial routers. Nevertheless the RSVP approach is not scalable. In fact the system resource requirements for supporting IntServ on routers increase in direct proportion to the number of the allocated RSVP sessions (i.e. number of application flows). Therefore, supporting a large number of RSVP reservations could introduce a significant negative impact on router performance. Hence, mainly due to its per-flow orientation, IntServ is viable within small-scale networks, but it is not suitable for large scale networks such as the high speed backbone and the Core Network.

2.2.2 Core Network

For the above reasons in the core network a scalable DiffServ based model is adopted. DiffServ [64, 17] is an alternative to IntServ that allows the deployment of different levels of best effort. The generic DiffServ deployment environment is based on the assumption that the network uses ingress traffic policing. Traffic entering the network, characterized by a traffic descriptor (TD), is passed through traffic shaping profile mechanisms, which bound average and peak rates, and set the packet class and discard criteria in accordance with the traffic profile and the SLA. DiffServ allocates network resources according to the packet class, allowing the high speed and high volume switching components of the network to operate without maintaining per-flow state information. In this environment, routers do not have to exchange explicit signalling messages to maintain states and to operate in a coordinate way. On the contrary, each router has only to apply a specific Per Hop Behaviour (PHB), which defines how the router must handle each received packet. Each PHB is executed locally in an independent way. In other words, a PHB refers to the packet scheduling, queuing, policing, or shaping behaviour of a node on any given packet belonging to a specific class. We just mention the main PHBs: Default PHB (basically Best Effort as defined in RFC 2474), Class-Selector PHB (as defined in RFC 2474), Assured Forwarding PHB (as defined in RFC 2597) and Expedited Forwarding PHB (as defined in RFC 2598). The cumulative behaviour of such stateless, local-context and distributed algorithms can yield the capability of supporting distinguished and predictable service levels in a scalable architecture. However DiffServ does not provide "hard QoS guarantees", it tunes traffic, but it does not avoid traffic congestion; the Qos experienced by the users depends again on the overall network dimensioning.

GRIP: a new DiffServ based QoS mechanism In order to improve the user perceived QoS, a novel solution called GRIP [15, 16, 87] is implemented in the core network. The GRIP protocol introduces the concept of of Admission Control. The main idea is to check the possibility to accept a new connection maintaining a suitable QoS of the ongoing connections. GRIP takes advantage from the cooperation of Endpoint Admission Control (EAC) [22] and Measured Based Admission Control (MBAC) techniques [38, 21]. In MBAC, each router measures resource consumption and the aggregate traffic that is handling. Admission Control decisions are then taken on such measurements, rather than on the basis of per flow state information. The driving

idea of EAC is that, upon connection set-up, each sender-receiver pair starts a Probing phase whose goal is to determine whether a new connection can be admitted to the network while maintaining the QoS requirements of the already admitted connections. To support GRIP, the routers must simply implement the priority queuing mechanisms that is a standard functionality of DiffServ routers.
The inter-operation between IntServ and enhanced DiffServ [13, 92] takes place within the Border Routers (i.e. the routers located at the interface betweeen the core and the edge networks)and requires the implementation of the IntServ/DiffServ service mapping and the subsequent activation of the GRIP admission control.

The main advantages of this hybrid IntServ-DiffServ architecture are twofold: 1)The overall complexity of the network is limited and the scalability is assured; 2) Backward compatibility with standard DiffServ routers; this allows a smooth migration path from the actual best-effort Internet to a DiffServ architecture and finally to a GRIP architecture.

2.2.3 Mobility

When IP routing was originally defined, mobility of hosts was not considered to be an issue. Routing methods were built for static networks; the IP address encodes the computers physical location, and, by default, the location is fixed. Mobile IP defines protocols and procedures by which packets can be routed to a mobile node, regardless of its current point-of-attachment to the Internet. The research on Mobile IP is based on the results produced by the IETF activity and reported on RFCs 1853, 2002-2006, 2344, and 2356. Packets destined to a mobile node are routed first to its home network, the network identified by the network prefix of the mobile node's (permanent) home address. At the home network, the mobile node's home agent intercepts such packets and tunnels them to the mobile node's most recently reported care-of address, namely the address of the network in which the mobile host is currently attached. At the endpoint of the tunnel, the inner packets are decapsulated and delivered to the mobile node. IPv6 significantly increases efficiency of this procedure, mainly for the following reasons:

- Mobile IP has to assign global IP addresses to a mobile node on each point it attaches to the Internet. Due to the address shortage in IPv4 there may be problems on some links to reserve enough global IPv4 addresses.

- Using stateless address autoconfiguration and neighbor discovery mechanisms Mobile IPv6 neither needs DHCP nor foreign links to configure the care-of addresses of mobile nodes.

- IPSec is a standard feature of IPv6.

- To avoid waste of bandwidth due to triangle routing, Mobile IP specifies the mechanism of Route Optimization. While Route Optimization is an additional functionality for Mobile IPv4, it is an integral part of Mobile IPv6. All the nodes of the network (both routers and hosts that implement IPv6) are aware that it is possible that a Node is assigned an address for identification (home address) and another one for localization (care-of address). Then it is possible to improve the overall performances of the network by sending packets directly to the address used for localization purposes instead of using the intermediate Home Agent for this job.

Mobility and QoS. This "standard" IP mobility architecture must be integrated with specific entities to allow the Mobile user to maintain suitable levels of QoS. We have seen in section 2.1.2 that we envisage two possible types of handover, *Smooth Handover* and *QoS-Aware Handover* [23]. In Smooth Handover, mobile users accept a momentary degradation of QoS during handover. This is due to the fact that, after establishing a new connection, the QoS on the new link is just best effort and remains in this mode until a new RSVP protocol interaction is begun by the applications. This allows the integration of mobile IP and QoS without requiring any specific upgrade of network entities.

QoS-Aware Handover has been studied and designed to provide a suitable QoS even during the handover, at the cost of the introduction in the network of new entities. In particular, we require to modify the current RSVP architecture. The mobile host, transmits RSVP packets on the new link in order to reserve enough resources for providing a suitable QoS and simultaneously transmit RSVP packets on the old link to keep alive the old ones. In other words, the main idea behind QoS-Aware Handover is that we can test and reserve the resources on the new link before we actually perform the Handover. This means that even during the Handover, the QoS-Aware Network Architecture is able to support the expected level of QoS by means of the old link. After the Handover the packets belonging to the QoS-Sessions will immediately find the needed resources on the new link.

Chapter 3

Downlink Scheduling for Multirate Wireless Networks

There is tremendous momentum in the wireless industry towards next generation (3G and beyond) systems. These systems will not only migrate the existing voice traffic to a higher bandwidth platform, but are also expected to jumpstart large scale data traffic. Next generation wireless systems are being designed, standardized, built and deployed aimed at realizing this vision.

These emerging wireless systems such as CDMA, wideband OFDM and multislot TDMA allow multiple codes (channels) to be allocated to users, in each of which traffic can flow in one of multiple rates. This provides them more flexibility than is available in current systems to manage and modulate the traffic. This also gives rise to novel scheduling problems that we study in this section.

Our contributions are threefold.

- We abstract a general downlink scheduling problem which has many novelties. For example, we embody channel characteristics guided by communication theoretic considerations, and the properties of these channels get exploited in our scheduling algorithms. Second, we study QoS parameters related to per request behavior, in particular, we focus on optimizing response time per request. In contrast, prior work in wireless systems scheduling has typically focused on rate optimization metrics.

- The scheduling problems that arise above are hard to solve exactly since we show them to be NP-complete. However, we use an unusual analysis technique: *resource augmented competitive analysis*, to derive simple, online algorithms which are not only practical, but also provably have good performance in approximating the optimal maximum response time of a job.

- We present a detailed experimental study of our algorithms. Using real web server request logs and realistic 3G/4G system parameters, we show experimentally that our online algorithms perform significantly better than our worst-case analyses indicate.

This work combines aspects of combinatorial optimization (convex programming), scheduling algorithmics (analyzing online algorithms with augmented resources), and applies them to general scheduling problems that arise in next generation wireless systems. As our results indicate, the proposed scheduling algorithms can pack power and codes effectively, that is, they benefit from the multiple code, multi-rate feature of 3G/4G systems.

The rest of the chapter is organized as follows. We present an overview of the wireless network and channel characteristics, and abstract our scheduling problem in Section 3.1. In Section 3.2, we present a theoretical study showing the structure and challenge in these problems. In Section 3.3, we present our main algorithmic results, namely, simple online algorithms and present our augmented resource based analyses. In Section 3.4, we present our experimental results, and conclude with related work in Section 3.5.

3.1 Problem Formulation

In this section, we will describe the context of next generation wireless systems where scheduling problems arise, and abstract them for further study.

3.1.1 Wireless Network Model

We assume a packet cellular architecture in which each cell has a base station and is connected by a high-speed backbone to the Internet. Cells could be partially overlapping, however, neighboring cells coordinate on resolving conflicts with resource usage and interference. Each base station handles all requests to and from mobile users within the cell, i.e., it handles both uplink (from mobile users) and downlink (to mobile users) requests. Our focus here is on the downlink channel performance, which is likely to be a major focus in emerging systems since data traffic

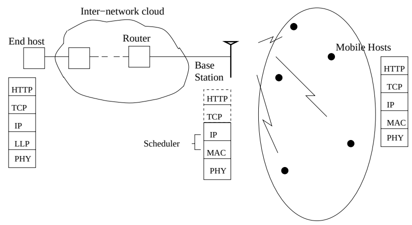


Figure 3.1: Scheduling scenario.

is expected to dominate over time and data traffic typically tends to have asymetrically large downlink demand.

Providing consistent quality of service to mobile users in the downlink is clearly important. However, wireline scheduling and resource allocation algorithms can not be directly applied to manage the downlink. Wireless networks have unique characteristics, an example being location dependent channel errors. Users in different regions of a cell experience different error rates, and hence they may get different data rates on the downlink. Unlike traditional scheduling scenarios, in a wireless environment, the scheduler must consider channel state in order to provide reasonable QoS, and wireless systems have a variety of built-in capabilities to gather channel condition information for this purpose.

In addition, random channel errors may result in poor performance of transport protocols such as TCP. Link layer retransmissions do not help, but aggravate the situation since they interfere with TCP's rtt computations [77]. Most solutions to this problem propose intercepting the connection at the base station, creating two logical connections [77, 4]. The base station thus acts as a proxy, and interprets the packets up to the transport layer in order to address random channel errors. Proxy servers store-and-forward data to mobile users, and thus have information regarding the various requests in the system. We will assume this context and address the scheduling problems that arise.

3.1.2 Communication Channel Model

In wireless systems, channels have variable attenuation depending on the geographic location of the users. This is mainly due to multipath impairments and radio propagation losses. Say the base station (BS) is communicating with n mobile users. The physical channel attenuations of the users are denoted by $\bar{g}_1, \bar{g}_2, \cdots, \bar{g}_n$ respectively; each $\bar{g}_i$ is a scalar factor which we call the *physical* gain. If the BS transmits power p_i to a user i, the signal-to-interference-plus-noise ratio (SINR) is given by $SINR = \frac{\bar{g}_i p_i}{\sigma^2}$, where σ^2 is the total noise power (including interference) [71]. SINR determines the rate of transmission of packets to the user[1]. In particular, the rate $r(\cdot)$ as a function of the SINR is a concave logarithmic function [25, 27],

$$r_{bps}(x) = \bar{W} \log_2(1 + \frac{x}{\Gamma}) \tag{3.1}$$

where $r_{bps}(\cdot)$ is the rate in bits per second, $\bar{W}$ is the spectral bandwidth used, and Γ is dependent on the coding gain from the physical layer error-correcting code [25]. Both Γ and $\bar{W}$ are system parameters, which for our purposes will be constants. Therefore, for a particular user, the service received over a period of time τ, obtained as a function of the power p_i allocated to the user on a single channel (code), is given by

$$r_i(p_i) = \bar{W}\tau \log_2(1 + \frac{\bar{g}_i p_i}{\sigma^2 \Gamma}) \tag{3.2}$$

The rate vs. SINR curves for next-generation wireless systems closely approximate the convex function described by this equation (see for example [12]). Figure 3.2 shows an instance of rates from measurements and from the equation above [12]. Note that the rate function is not linear in its argument and it is in fact concave. This rate function already embodies the effect of variable rate error-correcting coding schemes in the physical layer, as is typical in next generation wireless systems [25, 12, 63]. Therefore, we will use this equation for rate calculations in our scheduling problems. For notational convenience we will denote $g_i = \frac{\bar{g}_i}{\Gamma \sigma^2}$ as the *channel gain* and we will set $W = \tau \bar{W}$ yielding $r_i(p_i) = W \log_2(1 + p_i g_i)$.

3.1.3 Overview of Next Generation Wireless Multirate Data Networks

Our scheduling model abstracts multirate scheduling in many next generation wireless data systems. Here, we will briefly review two examples, namely, CDMA and and OFDM systems, and leave other examples such as multislot TDMA (EDGE) out of the discussion.

CDMA In *code division multiple access* (CDMA) systems [86], all the users share the entire transmission bandwidth and users are distinguished by the use of *signatures* (also called *codes*)

[1]The SINR is an important parameter for two reasons. This determines the probability of error in transmission of packets. Also, for a given error probability, we can transmit at higher rates dependent on the SINR. For example, when we have a higher SINR, we can transmit at a higher information rate for the same error probability [71].

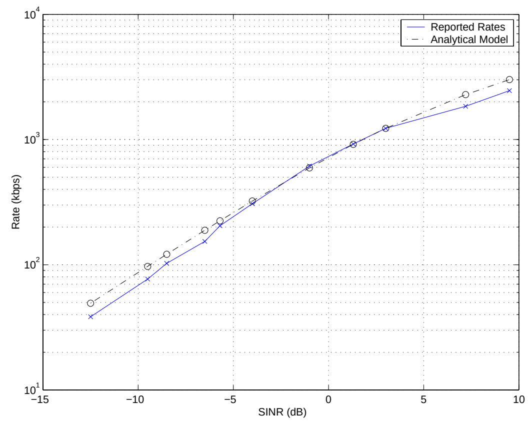


Figure 3.2: Data rates as a function of SINR: x-axis is on a logarithmic scale of decibels (dB) defined as $10\log_{10}(X)$.

assigned to them. There are two main CDMA proposals competing for the next generation standards. One is *cdma2000* which is an evolution of Qualcomm's IS-95 second generation CDMA system and is designed to use a frequency spectral bandwidth of 1.25MHz. The other is the *wideband CDMA* (WCDMA) which uses a larger frequency spectral bandwidth of 5 MHz and is being specified in Europe and Japan [63]. Both cdma2000 and WCDMA envisage higher rates through assigning multiple codes to users (*code-aggregation*) and variable rates through coding techniques. These two proposals are intended for both data and voice applications. Another system of significant interest is a purely data system called the *High Data Rate* (HDR) system being designed by Qualcomm [12]. This system is designed with a large range of available data rates, using sophisticated error-correcting coding schemes with higher latency, making it suitable for non-real time data traffic. In all CDMA proposals, there is a pilot signal in the broadcast control channel that enables the access terminal to measure the link channel conditions and this is reported back to the base-station. Therefore, the base-station has access to transmission conditions for users typically at a time-scale of a few milliseconds per measurement.

We briefly describe the physical, networking and systems issues related to the cdma2000 system, though similar issues are addressed in both the HDR and the WCDMA systems. A physical bandwidth of 1.2288 MHz is occupied by the transmission signal and 16 orthogonal (non-interfering) codes are used by the system. The physical layer has turbo codes [14] which are near optimal error-correcting codes, operating close to the fundamental channel rate limits

[27]. The system only allows certain discrete set of rates i.e., {38.4, 76.8, 102.6, 153.6, 204.8, 307.2, 614.4, 921.6, 1228.8, 1843.2, 2457.6} kilobits per second (kbps) over a time-frame of length 1.67 milliseconds. (These discrete rates translate to discrete allowable power assignments.) This results in $rate$(kbps)$\times$ 1.67 aggregate number of bits per time-slot, if all the codes are given to a single user. *Note that the successive rates available are at most a factor of two apart*, a fact we will use later.

The wireless data system is designed to have large geographic area coverage supporting a range of mobilities among the users. However, due to limitations in coverage are and interference management, the geographical areas are broken into cells. Each cell is served by a single base-station (or wireless access point). Several base-stations together could be connected to a wireless access network such as an Master Service Station, which serves as a conduit to the internet. The neighboring cells are typically uncoordinated and therefore their signals are treated as interference to the cell of interest. Interference management techniques are usually used, such as transmitting only short control channel information while idling, *i.e.* when users' queues are empty. Also, frequency planning and signal processing methods are also used for interference management [71].

Wideband OFDM Another air-interface which is being actively studied is based on wideband *Orthogonal Frequency Division Multiplexing* (OFDM) [24]. Here the wideband channel is divided into narrow frequency *tones* in a manner similar to traditional frequency-division multiplexing. However, there are important differences since the transmission frame is assembled using a Fast Fourier Transform (FFT) in OFDMs. As a result, transmissions become less susceptable to multipath propagation effects [24]. Also, the use of FFT allows one to dynamically assign different sets of tones to different user. Moreover, using powerful error-correcting codes variable rates can allocated on the tones. Though this is not among the third generation wireless standard proposals, it is receiving significant research and industry attention. The idea of OFDM is already a part of European digital audio broadcast standards the US wireline DSL standards [82], the HiperLan wireless LAN standard [88] and several wireless local loop systems. [18]. Systems with number of tones ranging from 16 to 256 are being currently studied [24].

3.1.4 Abstract Scheduling Problem

In this section, we abstract a general scheduling problem that arises in multirate, multi-code next generation wireless data systems such as the ones above. Our focus is on downlink and non-real time traffic (such as browsing, downloads, etc.).

Time is assumed to be partitioned into equal width windows called *time slots*[2] (or frames), whose width is τ. The base station has a total power P to transmit in a time slot. We also assume that there are a total of C codes which can be assigned to users in a time slot. If user u makes a request i, then we say that the gain of request i is the same as the gain of the user u, $g_i = g_u$. Requests [3] arrive in the system over time at the beginning of time slots. The size s_i (say in bits) and the channel gain g_i of the user who made the ith request are known when the request arrives at time a_i. The arrival time is also known as *release time.* We will assume that the channel conditions of the users are constant over the scheduling period. Although this is a simplification, it holds in some significant cases[4] and additionally, this already proves challenging. We leave the more general problem of time-varying user gains for future study.

The scheduling problem is to determine an assignment of power and codes to each user in each time slot, that is, to determine $\mathcal{C}_u(t)$, the set of codes assigned to user u at time t, and $p_u^{(i)}(t)$, the power assigned to user u at time t to each code $i \in \mathcal{C}_u(t)$. This translates to effective rate per code as given by Equation (3.2). The assignment must satisfy the following conditions.

- *Discrete Rate Set* : Only a discrete set of rates (equivalently, minimum power per discrete rate) is allowed. These rates denoted $R(1), R(2), \ldots$ have the property[5] that $\frac{R(i)}{R(i-1)} \leq 2$.

- *Request Completion*: All requests get the requested data size, that is, we need

$$s_u = \sum_t R_u(t) = \sum_t \sum_{i \in \mathcal{C}_u(t)} r_i(p_i) \tag{3.3}$$

where $\mathcal{C}_u(t)$ is the set of codes assigned to user u in time slot t, and $r_i(p_i)$ is calculated using (3.2) for the continuous or discrete cases as needed.

[2]We eill use time and time slot interchangeably when no confusion arises.

[3]The terms: requests, jobs and users, will be used interchangeably

[4]If time scale of the scheduler is several seconds, and if the users do not have very high mobility, then the channel conditions will be static over this time scale [71].

[5]This relationship holds for existing next generation wireless data system proposals like cdma2000 and HDR, which has a rate set of {38.4, 76.8, 102.6, 153.6, 204.8, 307.2, 614.4, 921.6, 1228.8, 1843.2, 2457.6} kilobits per second (kbps). The factor 2 is not sacrosanct. If the discrete rates are more spread out, but bounded by some constant, all our results will apply with minor changes in the claimed bounds.

QoS Metric

Various quality of service metrics could be optimized. We focus on one metric, namely response time[6] and our criterion is to *minimize the maximum response time*, where response time for request i is $c_i - a_i$, if request i is completed by time c_i. Although our focus is on this metric, some of our algorithms could be adapted for other metrics, such as *minimize total weighted response time*, namely $\sum_i w_i(c_1 - a_i)$, where arbitrary weights w_i are specified for each request i. If the weight for each job were 1, we have the traditional *average response time* measure. If the $w_i \propto 1/t_i$ where t_i is the time it takes to service the ith request in a completely unloaded system, that is, when all codes and power assigned to request i. This measure, $\frac{c_i - a_i}{t_i}$, is known as *stretch* of a job and it has been used in web server scheduling context for heterogeneous load [60, 61, 62].

We assume requests may be served over several time slots with different sets of codes at each time slot. In standard scheduling terminology [60], this corresponds to requests being *preempted* (*i.e.*, stop processing a request, process other requests, and resume the original request) or being *migrated* (*i.e.*, assign sets of codes to a user that differ from one time slot to another).[7]

Thus, our goal is *to come up with a schedule that minimizes the maximum response time, given an instance of jobs.* There are two basic variants of our problems, namely *offline* or *online.*

Offline problem : All request arrivals are known ahead of time. The offline case is of theoretical interest and is mainly useful to quantify the benefit to be accrued from scheduling.

Online problem : Requests arrive over time and the scheduling algorithms have to take their decisions without knowledge of future requests. The performance of the online algorithms are measured in comparison to the offline case.

We will consider variations of our scheduling problems by allowing rate (power) to take on any value in a range rather than restricting it to a discrete set; we call this the *continuous* version of our general problem above which has discrete rate (power) sets. This version proves very useful for developing the intuition that leads to our scheduling algorithms.

[6]This is also sometimes called flow time in literature.

[7]A more detailed model may distinguish some codes to be more preferable than the others from one time slot to another to insure intercell interference avoidance, an issue we do not consider in this paper.

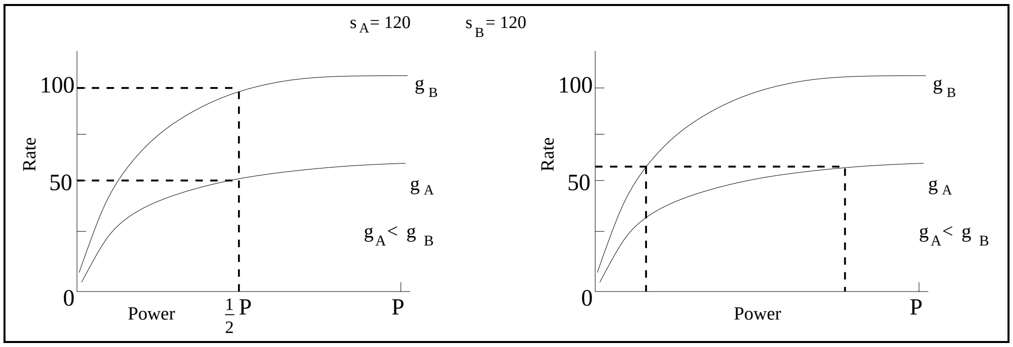


Figure 3.3: Example of non-trivial scheduling.

3.2 Understanding the Scheduling Problems

We start providing an example that shows that the assignment of codes and powers in our model is non-trivial. Assume there are two requests q and r with gains g_q and g_r respectively, and $a_q = a_r$. Let $g_q < g_r$. Let the sizes of both the jobs be $s_q = s_r = 120$. There are two codes available in the system with total power P. Consider the simple code and power allocation scheme in which all the power and codes are given to one request at a time. This is illustrated in Figure 3.3(a). Request q gets rate $R_q = 200$ in each frame, hence it takes 1 frame to complete. Request r gets rate $R_r = 100$ in each frame, hence it takes 2 frames to complete. If request q is served before request r, then $c_r = 3$ and thus the maximum response time is 3 frames. Consider the optimal schedule shown in Figure 3.3(b). In this schedule, in each time frame, request q is given one code and power sufficient to obtain 60 units of data, while request r is given the other code and the remaining power, which in this case, serves 60 units. This schedule will complete both the requests in just two time slots giving maximum response time of 2. Observe that the same arguments can be used to show that schedule shown in Figure 3.3(b) is optimal for this input also to minimizing total/average response time. A different example can be generated much like this one, but with the strategy in Figure 3.3(a) being the optimal one. Hence, we can conclude that the optimal schedule is not simply one of these two extreme schedules, but it is highly dependent on the input data.

3.2.1 Some Structural Observations

Here, we state and prove some properties of the communication channel. These properties serve to expose some aspects of the scheduling problems, but they have been helpful to us in designing scheduling algorithms, and later they will be invoked improving our main results.

Continuous Power (Rate) Case First, we consider the continuous power(rate) case, namely the case where the rates are not discrete and are a monotonic function of power, as in Equation (3.2).Concavity of the rate with respect to p in (3.2) implies that if we assign $c \geq 1$ codes to a user u, then *it is optimal to divide the total power p allocated to that user equally among the codes assigned.*

Lemma 3.2.1 ***(Equipartition of power)*** *Given c codes and power p to a user, the rate r is maximized for* $p_i = p/c,\ i = 1, \ldots, c$.

Proof: Say the user has physical gain $\bar{g}$. The function $r(p) = W\log(1 + \frac{\bar{g}p}{\Gamma\sigma^2})$ is concave in p. Hence,

$$\sum_{i=1}^{c} \frac{1}{c} r(p_i) \leq r(\frac{\sum_{i=1}^{c} p_i}{c}) \tag{3.4}$$

due to Jensen's inequality which states that $E[f(X)] \leq f(E[X])$ for a concave function $f(\cdot)$ [27].

■

Using lemma 3.2.1, if we have assign equal power among the codes allocated to the user, we can write the rate obtained by a user given c codes and p total power as,

$$R(p, c) = Wc\log(1 + \frac{gp}{c}). \tag{3.5}$$

Discrete Power (Rate) Case When we have a discrete rate set, the actual rate obtained on each code is given by the highest discrete rate[8] which is below $W\log(1 + \frac{gp}{c})$. Therefore, the difference between the continuous rate and the discrete rate case, depends on the discrete rate set available. Therefore, we can easily see the following fact.

Fact 3.2.1 *Denote by* $\{R(i)\}$ *the discrete allowable rates, with the property that* $\frac{R(i+1)}{R(i)} \leq 2,\ \forall i$. *If for the continuous rate problem* $R(k) \leq r_i(p) \leq R(k+1)$, *with* $r_i(p)$ *given by equation 3.2, then the following is true.*

$$\frac{1}{2}R(k) \leq \frac{1}{2}r_i(p) \leq R(k). \tag{3.6}$$

Proof: $R(k) \leq r_i(p) \leq R(k+1) \leq 2R(k) \Rightarrow \frac{1}{2}R(k) \leq \frac{1}{2}r_i(p) \leq R(k)$. ■

[8]Note that we make a regularity assumption that the user gains are such that the lowest discrete rate $R(1) \leq W\log(1 + gP)$, *i.e.* by allocating all the resources to the user there exists a feasible discrete rate. In practice, error-correcting codes can be used over a group of codes to increase the dynamic range of user gains that fall into the feasible region.

3.2.2 Computational Hardness.

In order to understand the challenge of the problem further, let us consider the offline complexity of the scheduling problems. We show that even in very restricted cases, this problem is NP-complete which means that efficient, that is, polynomial time algorithms exist if and only if $P = NP$, which is unlikely to be true [33]. If power (rate) values are required to be drawn from a discrete set, the problem in its simplest instance is the bin packing problem and hence it is NP-complete [33]. We focus on the more challenging case when code is discrete, as usual, but the power (and hence rate) is allowed to take any continuous value. In order to prove the hardness of this problem, we will consider the version of the problem in which ith request has arrival (release) time a_i, deadline d_i and size s_i in bytes. Each request must be completed within its deadline.

Theorem 3.2.2 *If the number of codes assigned to each user is integral, then it is NP-complete to compute a feasible schedule for the problem of meeting deadlines even if all users have the same channel gain, a common release time, a common deadline and the power assigned to each code is not restricted to a discrete set of values.*

Proof: The theorem is proved by reducing the NP-complete *3-partition* problem to the problem of meeting deadlines. The 3-partition problem is defined as follows [33]:

INSTANCE: A set A of $3m$ elements, $a_1, a_2, \ldots, a_{3m}$, a bound B and a size $s(a_j)$, $a_j \in A$, such that $\sum_{j=1}^{3m} s(a_j) = mB$.

QUESTION: Can A be partioned into m disjoint sets $A_1, A_2, \ldots, A_m$ such that for $1 \leq i \leq m$, A_i has three elements and $\sum_{a_j \in A_i} s(a_j) = B$?

The reduction is as follows: given an instance $\mathcal{I}$ of 3-partition we define an instance $\mathcal{J}$ of the combinatorial problem of meeting deadlines with $3m$ users. The request of user j, $1 \leq j \leq 3m$, has size $s_j = s(a_j)$; all requests are released at time 0 and have a common deadline $D = m$. All users have the same channel gain g and, therefore, the power that users need to get desired rates depends only on the number of codes they are assigned and on the size of the request. Let p_j denote the power assigned to user j if only one code is assigned to j over all frames. We assume that there are 3 codes per frame and that the maximum power of the base station is P where $\sum_{j=1}^{3m} p_j = mP$.

We now show that there is an assignment of users to frames that meets the common deadline D if and only if $\mathcal{I}$ has a feasible solution.

Assume that $A_1, A_2, \ldots, A_m$ is a feasible solution of $\mathcal{I}$. We define a solution of $\mathcal{J}$ as follows: if element a_j, $j = 1, 2, \ldots 3m$, is assigned to set A_i then user j is assigned to frame i with the minimum power that is needed to satisfy the user's request in one code in any one frame. It is easy to see that this is a feasible solution since $\sum_{a_j \in A_i} s(a_j) = B$ implies that P is the total power required by users assigned to frame i.

Similarly it is possible to show that given a feasible solution of $\mathcal{J}$ that satisfy all users' request within m frames, it is possible to obtain a feasible solution of $\mathcal{I}$. Namely, it is sufficient to assign to set A_i, $i = 1, 2, \ldots m$, the elements that correspond to users assigned to frame i; since the total power assigned to frame i is P it follows that $\sum_{a_j \in A_i} s(a_j) = B$. That completes the proof. ■

The result above in fact shows the problem to be NP-complete in the strong sense (see [33] for definition and significance). Although we have shown this hardness result only for the deadlines problem, it is easy to see that this immediately gives the hardness of other the scheduling problems we have, namely, minimizing the maximum and average (weighted) response times. Finally, notice that the result holds independently of how rates are affected by the use of multiple codes, since the hardness is proved even for the restrictive case when each request gets only one code over all time slots.

3.2.3 Offline scheduling problem

We study the offline version *i.e.*, when all arrival times are known apriori. Using this, we will get lower bounds on optimum values of certain QoS metrics which will be a benchmark to compare against online algorithms.

Deadlines Scheduling Problem We will focus on a particular variant of the problem, namely, that of meeting of deadlines. Here, each request j has an arrival time a_j as well as a deadline d_j. As before, for each request j, at time a_j we know its size s_j and the channel gain g_j The goal is to merely test feasibility, *i.e.*, determine if there is a valid schedule that meets all deadlines. This problem is the technical core of many other scheduling problems.

We can write the solution to the deadlines scheduling problem as a combinatorial optimization program as shown in Table 3.1. Here, $c(j, t)$ denotes the number of codes assigned to user

Time Indexed Program (integral)	Interval Indexed Program (fractional)
maximize 1	maximize 1
subject to	subject to
$\sum_{j=1}^{n} c(j,t) \leq C \; \forall t \quad \sum_{j=1}^{n} p(j,t) \leq P, \quad \forall t$	$\sum_{j=1}^{n} c(j,k) \leq C \; \forall k \quad \sum_{j=1}^{n} p(j,k) \leq P, \quad \forall k$
$\sum_{t=a_j}^{d_j-1} W c(j,t) \log\left(1+\frac{p(j,t)g_j}{c(j,t)}\right) \geq s_j, \; \forall j$	$\sum_{t=a_j}^{d_j-1} W \tau_k c(j,k) \log\left(1+\frac{p(j,k)g_j}{c(j,k)}\right) \geq s_j, \; \forall j$
$\sum_{t<a_j, t\geq d_j} c(j,t) = 0, \; \forall j$	$\sum_{k<a_j^{-1}, k>d_j^{-1}} c(j,k) = 0, \; \forall j$
$\sum_{t<a_j, t\geq d_j} p(j,t) = 0, \; \forall j$	$\sum_{k<a_j^{-1}, k>d_j^{-1}} p(j,k) = 0, \; \forall j$
$c(j,t), p(j,t)$ discrete values	$c(j,k), p(j,k) \geq 0, \; \forall j,k$

Table 3.1: Offline scheduling programs

j in time slot t and let $p(j,t)$ be the total power assigned to user j in time slot t over all the codes. This is called as the *time indexed program* in the table.

It is easy to see the following result.

Lemma 3.2.3 *The* integral time indexed program *has a feasible solution if and only if there exists a valid schedule for the deadlines problem in which all deadlines are met.*

This is a NP-hard problem as proved earlier, since $c()$ and $p()$ take on only discrete values. Hence, we relax the variables to be continuous and then show that the relaxed problem is tractable.

Fractional Time Indexed Program We relax the integer program by allowing $c(j,t)$ and $p(j,t)$ to be fractional.

Theorem 3.2.4 *There exists a psuedo-polynomial time algorithm to solve the Fractional Time Indexed Program.*

Proof:

We show that the program is convex and using previously known results, the theorem would follow. In order to show that the program is convex, we need to show that the constraint set is convex. Since all but rate constraints are linear, it is not obvious right away that the rate constraint is convex. This can be shown if $R_u(p,c) = Wc\log(1+\frac{gp}{c})$, defined in (3.5) is concave in (p,c). This is not obvious because, it needs to be shown that the function is jointly concave in both its arguments. However, the Hessian H for $R_u(p,c) = Wc\log(1+\frac{gp}{c})$ is given by

$$H = -\frac{Wg^2}{(c+gp)^2c}\begin{bmatrix} c \\ -p \end{bmatrix}\begin{bmatrix} c & -p \end{bmatrix}. \tag{3.7}$$

The Hessian is negative semidefinite, and therefore the function is concave (albeit not *strictly* concave) [58]. There exist polynomial time solutions for convex programming problems that test feasibility [58]. Using this on the program above, we obtain an algorithm for the deadlines scheduling problem that takes psuedo-polynomial time since the running time is polynomial not in input size ($n, \log s_i$ etc) but rather is polynomial in the number of variables which is bounded by the size of the numbers in the input (i.e., s_j).

■

Interval Indexed Program Though we have relaxed the integer programming problem to the pseudo-polynomial time algorithm, the problem size is still too big. In particular, the number of variables is $O(nT)$ where n is the number of requests and T is the total length of the schedule which could be large depending on request sizes (ideally, we would like to have the number of variables depend only on n, the number of requests). Next we consider decreasing the number of variables used in the convex program. We will define a new program below called the *interval indexed convex program.*

An *event* is either the arrival or the deadline of a request in the system. Consider the sorted list of the events $t_1, \ldots, t_K$. We divide the time scale into *intervals* where an interval is the time period between any two consecutive events, that is interval I_k contains $[t_k, t_{k+1})$. For n requests, the total number of intervals is at most $2n$. We will look for *sliver* solutions, that is, ones in which for each interval I, each user j gets power $p(j,t)$ and $c(j,t)$ for $t \in I$ that remains constant for all $t \in I$, that is, $p(j,t_1) = p(j,t_2)$ for $t_1, t_2 \in I$ and likewise for $c()$. Let $c(j,k)$ be the fractional number of codes and $p(j,k)$ be the fractional power assigned to job j in interval k *per time slot.* Let a_j^{-1} denote the interval at the beginning of which job j arrives in the system, and d_j^{-1} be the interval at the end of which its deadline lies.

The convex programming formulation for solving the scheduling problem with slivers is given in the table 3.1.

Theorem 3.2.5 *The time indexed convex program has a feasible solution if and only if the interval indexed convex program has a feasible solution. It can be solved in time polynomial in*

n, C using the convex program above.

Proof We will show that if the time indexed convex program has a feasible solution, so does the interval indexed convex program; the other direction is trivial.

Say $\overline{p(j,t)}$ and $\overline{c(j,t)}$ be the power and code assignments respectively at time frame t to user j in the time indexed convex program. Therefore, these values satisfy all the constraints of the time indexed convex program. We now claim that $p(j,k) = \frac{\sum_{t\in k}\overline{p(j,t)}}{\tau_k}$ and $c(j,k) = \frac{\sum_{t\in k}\overline{c(j,t)}}{\tau_k}$ are feasible values in the interval indexed convex program for user j in interval k, for all users and intervals. That is, these values would satisfy the constraints of the interval indexed convex program. Clearly we can bound $\sum_j p(j,k)$ as

$$\sum_j \frac{\sum_{t\in k}\overline{p(j,t)}}{\tau_k} = \sum_{t\in k}\frac{\sum_j \overline{p(j,t)}}{\tau_k} \leq \sum_{t\in k}\frac{P}{\tau_k} \leq P.$$

So the power constraint is satisfied; similarly, the code constraint is satisfied too. We have

$$\frac{1}{\tau_k}\sum_{t\in k}\overline{c(j,t)}\log(1+\frac{g_j\overline{p(j,t)}}{\overline{c(j,t)}}) \leq c(j,k)\log(1+\frac{g_j p(j,k)}{c(j,k)})$$

due to Jensen's inequality for multidimensional functions which states that $E[f(\mathbf{X})] \leq f(E[\mathbf{X}])$ for a concave function $f(\cdot)$ [27]. Therefore, the demand constraint is satisfied which completes the proof. ■

The result above exposes an interesting structural property of the interval indexed convex program, i.e., the structure that sliver assignment of power and code to requests is optimal in the fractional case.

Using the Deadlines Scheduling Problem Using the deadlines scheduling problem, other scheduling problems can be solved near optimally. For minimizing the maximum response time (max-flow), we would start by guessing a target response time F, and checking if there is a schedule in which the maximum flow for any request is at most F. This can be reformulated as the deadlines scheduling problem since for a job i to have flow at most F, it must have deadline $a_i + F$. which is the bound on the completion time, and therefore, is a deadline. Now our deadline scheduling problem can be used to check the feasibility of target F. If the target can not be met, we choose a larger value of F and continue. Else, we decrease the estimate F and continue. An efficient solution is to perform a binary search with the target flow value. That gives a efficient (polynomial time) algorithm to optimize the maximum response time. Indeed the same approach works for optimizing quality of service criteria such as $\max_i f(c_i - a_i)$ for any monotonic increasing function f.

3.3 Online Heuristics

In this section we present our main algorithmic results, namely, a set of online algorithms for optimizing the metrics of our interest. Recall that, at any instant, an online algorithm has to take all its scheduling decisions only on the basis of requests that were released in the past, without any assumption on the requests that will be presented in the future. We measure the performance of an online algorithm using a quite pessimistic measure. Namely, we consider the ratio between the value of the objective function achieved by the algorithm and the value obtained by an ideal adversary that knows the entire input sequence in advance and serves it optimally (there is no bound on the time the adversary needs to identify the optimal solution). This kind of analysis is widely known as Competitive Analysis of online algorithms [78].

In our analysis we also use a technique, known as *resource augmentation* [45]. That is, we compare the optimum (i.e. the solution found by adversary) with the value of the solution found by the online algorithm when it is provided with more more resources than the adversary. Formally, we say that algorithm A for our scheduling problem is an $(\alpha, \beta, \gamma, \delta)$ approximation if it provides a β approximation of the optimum when the sizes of user requests are scaled down by a factor α and the number of codes (the power) used by the algorithm is at most γ (δ, respectively) times the number of codes (the power, respectively) used by the optimum solution to serve the original input sequence.

This way of analysis may appear unusual: Why should one compare the performance of an algorithm against an adversary, if the algorithm is given more resources than the adversary (smaller request sizes, more codes, more power)? A first reason for this is conceptual. It is not very difficult to show that the scheduling problems we consider are not only hard to solve exactly as we did in Section 3.2 but, in fact, it is extremely hard to approximate the optimal solution (by choosing gain functions, arrival times, request sizes, etc. carefully) online. Given this negative scenario, resource augmentation based analysis provides a way to understand the inherent structure of the problem when simple, worst case analysis is of little help. A second reason concerns system design, since resource augmentation analysis gives an indication of the amount of extra resources needed in order to obtain a certain, guaranteed QoS. Namely, designers can provision the system with an amount X of resources and only guarantee the customer that (s)he can meet the best performance obtainable with a system that has an amount $X - \delta$ of resources, without revealing the actual capabilities of the system to the user. This may be an interesting guarantee in some cases. Finally, resource augmentation provides a tool to design new algorithms that perform well in practice, whereas worst case analysis would suggest that they have a poor behavior in theory. All these reasons apply to our work. We find new structure in the scheduling problems we study and we provide novel algorithms, which our experimental tests show to perform even better than our worst case analysis guarantees. Before proceeding with our results we give the following proposition.

Proposition 3.3.1 *Given an $(\alpha, \beta, \gamma, \delta)$ approximation algorithm A, it is possible to obtain a $(1, \beta, \alpha\gamma, \alpha\delta)$ approximation algorithm A'.*

Proof: Algorithm A' is as follows: first apply A to the given instance and let S be the obtained solution that allocates power and codes to users. For each slot $x \in S$, A' uses α copies of x and allocates these slots in the same way as x. Clearly the set of new slots allows to answer α times the demand satisfied by x. Notice that A' uses α times more codes and total power than A. This implies that all results stated for algorithms working on requests of reduced size can be transformed into results for algorithms working on the original instance of the problem at the expense of some extra codes and power allocated by the system. Therefore, we will not be concerned with directly forcing α to be 1 in our algorithms and their analysis, since no generality is lost in the process.

Minimizing the Maximum Response Time

It is well known in processor scheduling literature [60] that the online algorithm First In First Out (FIFO),also known as Earliest Release Time (ERT), is optimal for minimizing the maximum response time. Therefore, it is natural to ask how it would perform in our case. The simple FIFO strategy in our case allocates all the codes and power to one user at a time till the user completes the job. Using the "Equipartition of power" lemma (Lemma 3.2.1), this translates into giving the user a power per code of P/C in consecutive time slots that the user completely occupies. Therefore, we study the online scheme where each user is given a power P/C per code and is served by a FIFO scheduling discipline. We call this scheduling discipline FIFO($\frac{P}{C}$) and we first show a negative result, which demonstrates that such a strategy could be arbitrarily worse than the optimal strategy.

Theorem 3.3.1 *If the maximum response time of the scheduling discipline FIFO($\frac{P}{C}$), on an instance $\mathcal{I}$ of continuous job arrivals, is denoted by $f^{FIFO}(\mathcal{I})$ and the optimal discipline has a maximum response time of $f^{OPT}(\mathcal{I})$ then $\max_{\mathcal{I}} \left(f^{FIFO}(\mathcal{I})/f^{OPT}(\mathcal{I})\right) > M$ for any M.*

Proof: Let $C-1$ jobs arrive in a batch every time slot so that they need $\frac{P+\delta}{C}$ power each and one code to complete and such that $0 < \delta \leq \frac{P}{C-1}$, which implies that the jobs can be scheduled in one time slot.[9] Such a sequence would be scheduled in two time-slots by FIFO($\frac{P}{C}$), since it assigns two codes for each job, and therefore, would need $2C-2$ codes for every $C-1$ jobs. Hence, the job batch that arrives at the Mth time-slot is scheduled in the same time slot by the optimum and therefore has a response time of 1, whereas the online FIFO($\frac{P}{C}$) serves this job set only after $2(M-1)$ time slots have elapsed. As a consequence, each job in the set has a response time $2+2(M-1)-M = M$, showing that exists an instance $\mathcal{I}$ such that $\left(f^{FIFO}(\mathcal{I})/f^{OPT}(\mathcal{I})\right) > M$ for any M. ■

This problem arises because, for continuous arrivals, jobs may wait in the system for long periods of time.

In spite of the negative results above, we can show that FIFO($\frac{P}{C}$) is able to achieve the optimum if every request is reduced to $\frac{1}{2}$ of its original size. To do this we need the following lemma.

Lemma 3.3.2 *Let the optimal discipline on an online job arrivals instance $\mathcal{I}$ give a power assignment $p_j^{OPT}(\mathcal{I})$ and code assignment $k_j^{OPT}(\mathcal{I})$ to the jth job. Let us denote by $\mathcal{I}'$ the instance where each job size s_j in instance $\mathcal{I}$ is reduced to $s_j/2$, and denote by $k_j(\mathcal{I}')$ the code assignment to job j by the scheduling discipline FIFO($\frac{P}{C}$) applied to $\mathcal{I}'$. Then*

$$\frac{k_j(\mathcal{I}')}{C} \leq max\left\{\frac{p_j^{OPT}(\mathcal{I})}{P}, \frac{k_j^{OPT}(\mathcal{I})}{C}\right\} \tag{3.8}$$

Proof For each job j of size s_j on the original instance $\mathcal{I}$, let the pair p_j^{GS}, k_j^{GS} be such that

$$\left(p_j^{GS}, k_j^{GS}\right) = \text{argmin}_{(p_j^c, k_j^c): s_j \leq W k_j^c \log(1+\frac{g_j p_j^c}{k_j^c})} \left[\frac{p_j^c}{P} + \frac{k_j^c}{C}\right] \tag{3.9}$$

[9]Recall that the maximum power and codes available in a slot are P and C respectively.

where p_j^{GS}, k_j^{GS} are the total power and codes assigned to user j. This solution allocates a power per code $p_j^s = p_j^{GS}/k_j^{GS}$ to the jth job/user. We now show that allocating power P/C per code is not much worse in terms of this criterion. For each code which uses power p_j^s we allocate $r_j^s = \left\lceil \frac{p_j^s}{P/C} \right\rceil$ codes with power $\frac{P}{C}$ in each code. Since $r_j^s \geq \frac{p_j^s}{P/C}$,

$$\begin{aligned} s_j \overset{(a)}{\leq} W \log(1 + p_j^s g_j) &\leq W \log\left(1 + r_j^s \frac{P}{C} g_j\right) \\ &\leq W r_j^s \log\left(1 + \frac{P}{C} g_j\right). \end{aligned} \tag{3.10}$$

where (a) is due to (3.9). Hence using this allocation the demand s_j of each user j is satisfied. As a result we can give $k_j^{alloc} \overset{def}{=} k_j^{GS} \lceil \frac{p_j^{GS}/k_j^{GS}}{P/C} \rceil$ codes of power $\frac{P}{C}$ and still complete the job. Therefore, for the P/C allocation, if we use k_j^{alloc} codes for the job j, we obtain,

$$\frac{k_j^{alloc}}{C} = \frac{k_j^{GS}}{C} \left\lceil \frac{p_j^{GS}/k_j^{GS}}{P/C} \right\rceil \leq \left(\frac{p_j^{GS}}{P} + \frac{k_j^{GS}}{C} \right) \tag{3.11}$$

But we have $p_j^{alloc} = k_j^{alloc} P/C$, and hence,

$$\frac{1}{2}\left(\frac{p_j^{alloc}}{P} + \frac{k_j^{alloc}}{C} \right) = \frac{k_j^{alloc}}{C} \leq \left(\frac{p_j^{GS}}{P} + \frac{k_j^{GS}}{C} \right) \tag{3.12}$$

We now relate this to the optimal solution on the instance $\mathcal{I}$. Let us denote by $p_j(\mathcal{I})$ and $k_j(\mathcal{I})$ respectively the total power and the number of codes allocated by the P/C allocation when applied to the instance $\mathcal{I}$. Therefore[10],

$$\begin{aligned} \frac{k_j(\mathcal{I}')}{C} &\leq \frac{1}{2} \frac{k_j(\mathcal{I})}{C} \\ &\leq \frac{1}{2}\left(\frac{p_j^{GS}(\mathcal{I})}{P} + \frac{k_j^{GS}(\mathcal{I})}{C} \right) \\ &\leq \frac{1}{2}\left(\frac{p_j^{OPT}(\mathcal{I})}{P} + \frac{k_j^{OPT}(\mathcal{I})}{C} \right) \\ &\leq max \left\{ \frac{p_j^{OPT}(\mathcal{I})}{P}, \frac{k_j^{OPT}(\mathcal{I})}{C} \right\} \end{aligned} \tag{3.13}$$

■

[10]Note that the optimal assignment need not necessarily assign equal power per code across time slots it schedules the jth user. However, due to the joint concavity of the rate in terms of power and code assignment (see in proof of Theorem 3.2.4) the rate for a given user can only increase by giving equal power assignment per code across time-slots, provided the power constraint is satisfied. Therefore, the optimal solution has a tighter constraint than the minimization in (3.9) and hence the third inequality in (3.13) is satisfied.

Theorem 3.3.3 *Let the optimal discipline on an online job arrivals instance $\mathcal{I}$ have a maximum response time of $f^{OPT}(\mathcal{I})$. Let us denote by $\mathcal{I}'$ the instance where each job size s_i in instance $\mathcal{I}$ is reduced to $s_i/2$, and denote the maximum response time of the scheduling discipline FIFO($\frac{P}{C}$) applied to $\mathcal{I}'$ by $f^{FIFO}(\mathcal{I}')$ then*

$$f^{FIFO}(\mathcal{I}') \leq f^{OPT}(\mathcal{I}) + 2, \ \forall \mathcal{I}. \tag{3.14}$$

Proof: Consider the request r achieving the maximum flow time for the scheduling discipline FIFO($\frac{P}{C}$) applied to instance $\mathcal{I}'$. W.l.o.g., we assume that request r is the last request presented to the algorithm. Otherwise, to the purpose of our analysis, the sequence can be stopped when request r is released. Request r has been released in slot t_r and completed in slot $C_r(\mathcal{I}')$ in the algorithm's solution. Denote by t the last slot in which all requests that have been presented before time t have been completed by slot t. We can restrict our attention to the subset of user requests, denoted by $\mathcal{J}' = \{j \in \mathcal{J} | a_j \geq t\}$, that have been presented at or after slot t, since these are the only requests that contribute to the flow time of request r. The completion time for request r using the FIFO($\frac{P}{C}$) discipline on instance $\mathcal{I}'$ is at most

$$C_r(\mathcal{I}') \leq t + \left\lceil \frac{\sum_j k_j(\mathcal{I}')}{C} \right\rceil \tag{3.15}$$

where $k_j(\mathcal{I}')$ is the number of codes required to complete job j on instance $\mathcal{I}'$. Denote by s the user request completed last in the solution of the optimum on instance $\mathcal{I}$. Request s has been released at some time $t_s \leq t_r$ and therefore,

$$\begin{aligned} & C_s^{OPT}(\mathcal{I}) - t_s \geq t - t_s \\ & + max \left\{ \left\lceil \frac{\sum_j p_j^{OPT}(\mathcal{I})}{P} \right\rceil, \left\lceil \frac{\sum_j k_j^{OPT}(\mathcal{I})}{C} \right\rceil \right\}, \end{aligned} \tag{3.16}$$

Now, we have

$$\begin{aligned} & f^{FIFO}(\mathcal{I}') = C_r(\mathcal{I}') - t_r \\ \overset{(a)}{\leq} \quad & t - t_r + \left\lceil \frac{\sum_j k_j(\mathcal{I}')}{C} \right\rceil \\ \leq \quad & t - t_r + \frac{1}{2} \left\lceil \frac{\sum_j k_j(\mathcal{I})}{C} \right\rceil \\ \leq \quad & t - t_s + \frac{1}{2} \left(\left\lceil \frac{\sum_j p_j^{GS}(\mathcal{I})}{P} \right\rceil + \left\lceil \frac{\sum_j k_j^{GS}(\mathcal{I})}{C} \right\rceil \right) \\ \overset{(b)}{\leq} \quad & t - t_s + max \left\{ \left\lceil \frac{\sum_j p_j^{OPT}(\mathcal{I})}{P} \right\rceil, \left\lceil \frac{\sum_j k_j^{OPT}(\mathcal{I})}{C} \right\rceil \right\} + 2 \end{aligned} \tag{3.17}$$

$$\stackrel{(c)}{\leq} \quad C_s^{OPT}(\mathcal{I}) - t_s + 2 \leq f^{OPT}(\mathcal{I}) + 2,$$

where (a) follows from (3.15), (b) follows from Lemma 3.3.2 and (c) from (3.16), giving us the result.

■

This result shows that by reducing the demand, we can prove a positive result on FIFO$(\frac{P}{C})$. As seen in the following theorem (whose proof is similar to Theorem 3.3.3), if we allow for resource augmentation, a positive result can be shown on FIFO$(\frac{P}{C})$.

Theorem 3.3.4 *Let the optimal discipline on an online job arrivals instance $\mathcal{I}$ have a maximum response time of $f^{OPT}(\mathcal{I})$. Denote by $f^{FIFO_{Aug}}(\mathcal{I})$ the maximum response time of the scheduling discipline* FIFO$_{Aug}(\frac{P'}{C'})$ *applied to $\mathcal{I}$,where the power and the number of codes per time slot have been augmented to P' and C' respectively. Then there exists $P' \leq 2P$ and $C' \leq 2C$ such that,*

$$f^{FIFO_{Aug}}(\mathcal{I}) \leq f^{OPT}(\mathcal{I}), \ \forall \mathcal{I}. \tag{3.18}$$

Proof: Let the optimal scheme on instance $\mathcal{I}$ allocate power $p_j^l(i)$ to user j, on the ith code in time slot l. Let us assign $r_j^l(i) = \lceil \frac{p_j^l(i)}{P/C} \rceil$ codes with power P/C per code. Due to the power constraint we have for the lth time slot: $\sum_j \sum_{i=1}^C p_j^l(i) \leq P$. For every slot l, we can now easily give an upper bound to the total number of codes needed with the P/C allocation:

$$\sum_j \sum_{i=1}^{C} r_j^l(i) \leq \sum_j \sum_{i=1}^{C} \left\lceil \frac{p_j^l(i)}{P/C} \right\rceil + C \leq 2C, \tag{3.19}$$

where the second inequality is due to the power constraint. Therefore, there exists a P/C allocation which achieves the same schedule as the optimal but using power $P' \leq 2P$ and codes $C' \leq 2C$. The problem of scheduling jobs with P/C power per code is like a single processor scheduling problem [60], and FIFO is optimal for this problem with respect to maximum response time, proving the result. ■

3.3.1 The discrete case

In this section we show how to transform our algorithms for the continuous case into algorithms for the discrete case. Our transformation preserves the approximation of the algorithms in the continuous case at the expense of some extra codes and some extra power allocated in each slot.

In the continuous case we assign power P/C to every code, but this may correspond to a non-feasible transmission rate at the receiver for some specific user. To move from the continuous to the discrete case, we need to round the power assignment to a value that sustains one of the discrete transmission rates[11].

We will implement a rounding scheme that allows to turn a solution for the continuous case into a solution for the discrete case.

The algorithm for the discrete case works as follows:

1. Apply the continuous case algorithm on the sequence of job in input.

2. For each assignment of power z in a time slot, obtained applying the algorithm for the continuous case on the sequence of job in input, perform the following rounding scheme:

   (a) *Round up*: If there exists a power $\overline{z_1} \in (z, 2z]$ corresponding to a discrete rate, then assign power $\overline{z_1}$ to the code.

   (b) *Round down*: If there exists a power $\overline{z_2} \leq z$ corresponding to a discrete rate, then assign power $\overline{z_2}$ per code.

Then for each user we choose the rounding that gives the higher rate, if the user were given all the resources *i.e.*, all the power and codes. In the continuous scheme described in Subsection 3.3, all codes allocated to user j are assigned with power $x = P/C$ in the continuous solution.

Lemma 3.3.5 *The allocation scheme for the discrete case satisfies all users demands.*

Proof: User j is allocated with power per code x. For every code allocated with exactly x, rounding up will increase the transmission rate achieved on a code by a user. Rounding down will result, by Fact 3.2.1, in a transmission rate that is at least half of the transmission rate in the continuous case. Therefore, by assigning two codes, the transmission rate for the user does not decrease. Hence, the demand is satisfied by the rounding scheme. ■

Now, we show that the approximations shown for the continuous case can be translated to the discrete rate by additional resource augmentation.

[11]For the P/C allocation we impose a further regularity condition that the lowest discrete rate $R(1) \leq W \log(1 + \frac{gP}{C})$, *i.e.* there is a feasible discrete rate below this power allocation. Though this is perhaps a little more stringent than required, it makes the analysis simpler. As before this restriction can be removed in practice by using error-correcting codes on a group of codes, so that the combined rate is a feasible discrete rate.

Theorem 3.3.6 *Let the optimal discipline on an a continuous job arrival instance $\mathcal{I}$ have a maximum response time of $f^{OPT}(\mathcal{I})$. Let us denote by $\mathcal{I}'$ the instance where each job size s_i in instance $\mathcal{I}$ is reduced to $s_i/2$. Let the scheduling discipline* $\text{FIFO}_{disc}(\frac{P}{C})$ *be obtained by taking the discipline FIFO($\frac{P}{C}$) and applying the above rounding procedure. Finally, denote by $f^{FIFO_{disc}}(\mathcal{I}')$ the maximum response time of* $\text{FIFO}_{disc}(\frac{P'}{C'})$ *applied to $\mathcal{I}'$, where the power and number of codes per time slot have been augmented to P' and C' respectively. Then there exist $P' \leq 2P$, $C' \leq 2C$ such that,*

$$f^{FIFO_{disc}}(\mathcal{I}') \leq f^{OPT}(\mathcal{I}) + 2, \ \forall \mathcal{I}. \tag{3.20}$$

Proof: Let $\mathcal{K}_1$ ($\mathcal{K}_2$) denote the number of codes whose associated power was rounded up (respectively down) in a particular time slot l. Clearly, $|\mathcal{K}_1| + |\mathcal{K}_2| \leq C$. Let the power allocated on each code i after rounding be denoted by $p^{rnd}(i)$, and clearly $p^{rnd}(i) \leq 2P/C, i \in \mathcal{K}_1$ and $p^{rnd}(i) \leq P/C, i \in \mathcal{K}_2$. Now, suppose in each time-slot, for every code $i \in \mathcal{K}_2$ we assign two codes with power $p^{rnd}(i)$, and for each code $i \in \mathcal{K}_1$ (whose power was rounded up) we assign one code. Clearly such an allocation will meet the same demand as the continuous rate FIFO($\frac{P}{C}$) schedule. Hence the schedule is equivalent to the the continuous rate FIFO($\frac{P}{C}$) schedule. Therefore using this and Theorem 3.3.3 the result (3.20) can be obtained. The only question that remains is how much resource augmentation was done to obtain this. In the new allocation we have used $P' = \sum_{i \in \mathcal{K}_1} p^{rnd}(i) + 2 \sum_{i \in \mathcal{K}_2} p^{rnd}(i)$ total power and $C' = |\mathcal{K}_1| + 2|\mathcal{K}_2|$ total codes. But we have,

$$P' \leq 2\tfrac{P}{C}|\mathcal{K}_1| + \tfrac{P}{C}2|\mathcal{K}_2| \leq 2P \tag{3.21}$$

$$C' = |\mathcal{K}_1| + 2|\mathcal{K}_2| \leq 2C.$$

Hence the new allocation uses at most a power $P' \leq 2P$ and a number of codes $C' \leq 2C$. ■

We can also extend the result in Theorem 3.3.4 to the discrete rate with more resource augmentation.

Theorem 3.3.7 *Let the optimal discipline on a continuous job arrivals instance $\mathcal{I}$ have a maximum response time of $f^{OPT}(\mathcal{I})$. Denote by $f^{FIFO_{disc}}(\mathcal{I})$ the maximum response time of the scheduling discipline* $\text{FIFO}_{disc}(\frac{P'}{C'})$ *when applied to $\mathcal{I}$ where the power and the number of codes per time slot have been augmented to P' and C' respectively. Then there exists $P' \leq 4P$ and $C' \leq 4C$ such that,*

$$f^{FIFO_{disc}}(\mathcal{I}) \leq f^{OPT}(\mathcal{I}), \ \forall \mathcal{I}. \tag{3.22}$$

Proof: We sketch the proof. Using the same argument of the proof of Theorem3.3.6 we can prove that using $P' \leq 4P$ power and $C' \leq 4C$ codes, the discrete algorithm can meet the same demand as the continuous rate $FIFO_{Aug}(\mathcal{I})$. Therefore using Theorem 3.3.4 the result (3.22) can be obtained. ■

Other optimization criteria

Although we focused our attention on minimizing the maximum response time, several of our ideas could be extended to other optimization criteria such as *minimize total weighted response time*, $\sum_i w_i(c_1 - a_i)$, where arbitrary weights w_i are specified for each request i. If $w_i \propto 1/t_i$, where t_i is the time it takes to service the ith request when all codes and power are assigned to request i, the corresponding metric, i.e. $\frac{c_i - a_i}{t_i}$, is known as *stretch* of job i. Stretch has been used in web server scheduling context for heterogeneous load [60, 61, 62]. While response time is skewed towards large jobs, since jobs with large service times also tend to have large response time, the relative response metric is independent of size, resulting in more fairness for all job classes. Since data requests in the emerging data systems and applications would very likely be heterogeneous, relative response is an attractive metric to investigate. Other weight functions may also be useful, although the two above are most common and we will focus on those.

Minimizing the Average Flow Time

Average Flow Time measures the average time spent by each user request in the system between the frame of release and the frame of completion. Therefore, for a sequence of n user requests released over time, we seek for optimizing $\frac{1}{n}\sum_j(C_j - a_j)$. It is a well known result [3] that Shortest Remaining Processing Time (SRPT), namely the algorithm that at any time t schedules the pending request with minimum remaining processing time, is an optimal algorithm for the scheduling problem of minimizing the average flow time on a single machine if job preemption is allowed, i.e. the execution of a job can be interrupted and resumed later on the same machine.

We will establish the worst case performance of SRPT when the demand of every user is reduced to $\frac{1}{2(1+\epsilon)}$ of the original demand. As commented above in the section, a similar result can be obtained by providing the system with $2(1+\epsilon)$ times more power and codes than the optimum. In particular we will show that under this condition SRPT achieves the optimum average flow time.

Theorem 3.3.8 *Algorithm SRPT achieves the optimum average flow time if every user demand is reduced to* $\frac{1}{2(1+\epsilon)}$ *of the original demand.*

Proof: Consider user j and denote by f_j^{red} and by f_j^{OPT} the number of frames used by the algorithm working with reduced demands for user j and a lower bound on the number of frames used by the optimum.

From Equation 3.13 it follows

$$f_j^{red} \leq \frac{p_j^r}{P} + \frac{k_j^r}{C} \quad \leq \quad max\{\frac{p_j^{OPT}}{P}, \frac{k_j^{OPT}}{C}\} \leq f_j^{OPT}$$

We know that SRPT on allocation f_j^{OPT} gives an optimal solution.

Consider the schedule produced by SRPT on f_j^{OPT} and stop processing request j when request j is allocated for f_j^{red} frames. This new schedule has an average flow time certainly smaller than SRPT on f_j^{OPT}. On the other hand, this schedule has an average flow time that is certainly higher than the result of applying SRPT on f_j^{red} for which the theorem follows. ■

Minimizing Weighted Response Time

We now show a P/C power/code allocation that optimizes the weighted response time of a set of user requests released over time. Recall that when the objective function to minimize is weighted response time, every job has an associated weight w_j. The goal is to optimize $\sum_j w_j(C_j - a_j)$. Observe that average stretch and average flow time are special cases of weighted flow time when respectively $w_j = \frac{1}{s_j}$ or $w_j = 1$. Finally, minimizing average stretch or average flow time is clearly equivalent to minimizing the total stretch or the total flow time respectively. The algorithm we propose is Highest Density first (HDF), that at any time t schedules the pending request with maximum ratio w_j/s_j, which we henceforth define *density*. The scheduling of a user request is preempted in a frame if a request with higher density is released. When a user request is completed, the pending request with highest density is scheduled.

Theorem 3.3.9 *For any* $\epsilon > 0$, *Highest Density first is an* $\frac{1+\epsilon}{\epsilon}$ *approximation for minimizing the weighted flow time if every request is guaranteed for a fraction* $\frac{1}{2(1+\epsilon)}$ *of the original demand.*

Proof: We denote by f_j^{red} and by f_j^{OPT} respectively the maximum number of frames used by the algorithm working with reduced demands and a lower bound on the number of frames used by the optimum. Since demand is reduced to $\frac{1}{2(1+\epsilon)}$ of its original size, equation (3.13) implies that $f_j^{red} \leq \frac{1}{1+\epsilon} f_j^{OPT}$. The optimum has to allocate at least f_j^{OPT} frames to request j in order to meet its demand.

For the sake of analysis, we compare HDF with a lower bound on the optimum given by a fractional version of HDF, denoted in the following by FHDF. HDF and FHDF work the same way, but FHDF is able to reduce the weight of a job fractionally, as the job is processed. In particular, if an amount s of job j has already been scheduled by frame t, then the weight of job j is reduced to $w_j^{FHDF}(t) = w_j(s_j - s)/s_j$. The size of job j at time t is analogously reduced to $s_j(t) = s_j - s$. Observe $w_j(t)/s_j(t) = w_j/s_j$, hence the density of a job remains constant along the execution of FHDF. This implies that FHDF schedules jobs in the same order as HDF.

The maximum ratio between the weights not completed by HDF and FHDF at any time t is an upper bound on the approximation ratio of the two algorithms. Since $f_j^{red} \leq \frac{1}{1+\epsilon} f_j^{OPT}$, it is easy to show that the number of frames allocated by FHDF to any request j by time t is never larger than the number of frames allocated to request j by HDF. It follows that the jobs that are completed by $FHDF$ at any time t are also completed by HDF within the same time. As a consequence, the ratio between the uncompleted weight of the two algorithms is given by the maximum over all jobs j of $\frac{w_j}{w_j^{FHDF}(t)}$. Assume the worst case in which both HDF and FHDF started job j at the same time. When HDF is about to complete request j, FHDF is still left with at least $\frac{\epsilon}{1+\epsilon} f_j^{OPT}$ frames to be allocated to request j, therefore with a fraction at least $\frac{\epsilon}{1+\epsilon}$ of the original weight of request j. The ratio between the remaining weight of HDF and FHDF is then bounded by

$$\frac{w_j}{w_j^{FHDF}(t)} \leq \frac{\epsilon}{1+\epsilon},$$

for which the claim of the theorem holds. ■

The Highest Density first heuristic, when specialized to average stretch and average response time, becomes the Shortest Processing Time first heuristic (SPT), that at any time schedules the pending request that has shortest processing time. (Note that the selection policy still applies to the SPT algorithm.)

Corollary 3.3.10 *For any $\epsilon > 0$, Shortest Processing Time first is an $\frac{1+\epsilon}{\epsilon}$ approximation for minimizing the average stretch (and response time) if every request is guaranteed for a fraction $\frac{1}{2(1+\epsilon)}$ of the original demand.*

3.4 Simulation Study

In this section, we study the performance of our online and offline algorithms experimentally. The results presented in this section are derived from real datasets from web logs and from synthetic datasets designed to explore certain features of the algorithms.

Online Algorithms The $FIFO(\frac{P}{C})$ algorithm was described in Section 3.3. We call this algorithm *FIFO-continuous*. Essentially, this algorithm allots P/C power to each code, and job requests are then scheduled in the order of their arrival.

The rounding procedure for converting the continuous power (rate) algorithm to a discrete power (rate) algorithm was described earlier in Section 3.3 3.3.1. Rounding the rates results in a different power per code for each job than in the continuous case. As a result, when codes are assigned to a job in a slot, the packing may not be tight. In other words, some power and/or codes might be unused in a slot. The goal of a discrete-rate online algorithm is to minimize this potential waste of resources in order to reduce the maximum response time.

With this goal in mind, we have developed three online discrete-rate algorithms, which we call *FIFO*, *2D-FIFO*, and *2D-PIKI*. Given a job, the power per code corresponding to the discrete bit rate is the same for all of these algorithms. They differ only in the way the jobs are selected for receiving service.

- **FIFO:** The request i currently in the system that has the earliest release time a_i is selected. No other job in the system is scheduled until this job is completed. This is the traditional FIFO algorithm.

- **2D-FIFO:** The request i currently in the system that has the earliest release time a_i has higher priority over other job requests. However, if this job i leaves power/codes unused in that time-slot, other jobs j in the system are considered in the non-decreasing order of their release times a_j. FIFO is the focus of our theoretical analysis while 2D-FIFO (and the following 2D-PIKI) allows us to estimate the performance decrease of FIFO due to its worst use of resources in the discrete case.

- **2D-PIKI:** The request i currently in the system that has the highest value of power per code p_i is selected for scheduling. If this job leaves power/codes unused in that time-slot, other jobs j in the system are considered in the non-increasing order of the power per code

p_j. This scheme aims to achieve a better packing in each time slot, in order to reduce the completion time.

Due to discrete nature of the rate set, in certain slots FIFO may have some codes k^{extra} and some power p^{extra} that cannot be assigned to any job in the system, since the power per code $p_i > p^{extra}$ for all jobs i. In such a situation, another scheduler (in particular 2D-FIFO or 2D-PIKI) will choose the first flow that received service in the slot and give it the best possible discrete rate with the remaining power and codes.

Note that no algorithm guarantees that all the power and codes will be used in every slot. Therefore, we expect to see differences between the *FIFO-continuous* algorithm and the three discrete-rate algorithms. In the remainder of this section, we will quantify the differences through simulations.

Channel Specifications We adopt the channel specifications similar to 3G system proposals [12, 63] for our channel model.[12]

We perform experiments on a single cell and abstract the effect of out-of-cell interferers into a decrease in SINR values. The peak power available at the base station was chosen to be $P = 40W$, while the maximum number of channels was chosen as $C = 16$. The power attenuation factor $\bar{g}_u$ for user u is modeled with two components: (a) shadow loss component S, which is a log-normal shadowing variable, and (b) path loss components $P = 1/d^{\alpha}$, where d is the distance between the base station and the user and α is the distance loss exponent. We chose $\alpha = 3$, giving $\bar{g}_u \propto S/d_u^3$.

The parameters to be used for the rate calculation given in Equation (3.2) were chosen as follows: $\tau = 1.67$ milliseconds, $\bar{W} = 76.8KHz$ and $\Gamma = 4.7dB$. We operated over an SINR range from -15dB/Hz to 15dB/Hz. The discrete rate set used is a set of 15 rates : { 2.4, 4.8, 9.6, 19.2, 38.4, 76.8, 102.6, 153.6, 204.8, 307.2, 614.4, 921.6, 1228.8, 1843.2, 2457.6 } Kbps. Under these restrictions, the maximum data rate for a mobile user in the cell will range from 10 Kbps to 2 Mbps.

Data Sets We used web-traces from DEC for generating large representative workloads. The traces used in the experiments are derived from a single proxy server. For comparing the online

[12] We would like to emphasize that our algorithms are applicable to all systems that support multiple channels and multiple rates. Such systems include the various next-generation wireless data networks.

algorithms with the offline optimum, we used traces that consist of up to 100 jobs that arrive over a period of 1 minute. To evaluate the performance of various online algorithms under heavy demand, we use traces consisting of 1000 jobs arriving over a period of 15 minutes. These requests are generated by nearly 100 users in the cell. In all of the traces used, the minimum request size was 40 bytes, the maximum request size was 500 kilobytes, with mean request sizes ranging from 20 - 34 kilobytes. The average inter-arrival time of requests in the traces is 300 - 400 milliseconds.

Simulation Tools For optimizing the convex programs in the offline case, we use an optimization tool called *LOQO* [68], along with a front-end tool named *AMPL* [67]. developed at Bell Labs. LOQO is a program for solving smooth optimization problems, and uses an infeasible primal-dual interior-point method applied to a sequence of quadratic approximations to a given problem. AMPL is a popular tool used as an Interface Description Tool for many linear/non-linear optimization programs. Convex programming is a fairly expensive operation, and our experiments were often limited by the program running out of memory for moderate amount of variables (in the order of several hundreds). We used these runs only for benchmarking purposes. Our online algorithms were evaluated using a custom-built simulator.

3.4.1 Experiments

We performed three types of experiments to evaluate our algorithms. The first two experiments validate our theoretical results and demonstrate some interesting properties of the online algorithms. We use both synthetic datasets and web-log traces for these experiments. The third experiment was designed to measure the average-case performance of our algorithms.

Online Heuristics In this section, we will evaluate the different online algorithms and compare their performance against the offline optimal algorithm. We used small web traces, with 100 jobs arriving over a period of 1 minute, for computing the convex programming lower bound, which we denote by OPT, for max-flow. The job requests are for users who are distributed uniformly in the cell. We present the max-flow results for four such traces along with the results for the online heuristics in Part I of Table 3.3: all max-flow values are in terms of slots.

It can be seen that the online algorithms perform very close to the optimal, on the average. From the table, we also see that *2D-FIFO* performs the best among the three discrete-rate

Trace	Id	Size (in bits)	Max data rate (in Kbps)	p_i/k_i (discrete)
config2	1	124000	7502.72	0.5501
	2	400	24.091	4.001
	3	650	48.376	4.001
	4	650	97.524	4.001
	5	650	198.095	4.001
	6	1200	408.048	4.001
	7	2000	860.529	4.001
	8	35000	9000.895	4.001

Table 3.2: Online Heuristics: config2

Trace	OPT (in slots)	Continuous FIFO	Discrete FIFO	Discrete 2D-FIFO	Discrete 2D-PIKI
Part I: Web traces					
config3	109257	109891	120682	114054	114587
config4	50249	50637	55281	52263	59467
config5	36460	36725	40325	38540	46432
config6	16224	16254	17280	17210	24711
Part II : Anomalous behavior					
config2	35	637	2595	69	35

Table 3.3: Online Heuristics: Performance

algorithms and also that *2D-PIKI* performs the worst. In addition, the discrete algorithms always appear to perform worse than the continuous version.

While these inferences continue to hold true in most instances, as we will show in the subsequent examples, they are not always true. Consider the example set of jobs shown in Table 3.2. The trace *config2* consists of this set of 8 jobs arriving at the same time, every 35 slots, for 1040 time slots. The max-flow results for this trace are presented in Part II of Table 3.3. There are some interesting observations to be made from this example. The first observation is that *2D-PIKI* performs as good as the optimal algorithm, deviating from its usual poor behavior. Moreover, while one might expect the discrete-rate algorithms to do worse than the continuous case algorithms in all cases, in this particular example, the converse is true. *2D-FIFO* and *2D-PIKI* perform much better than the *FIFO-continuous* algorithm.

Practical Scenarios In this experiment, we consider several arrangements of 100 users in a cell, and evaluate the impact on the performance of the algorithms. In particular, we consider three scenarios: (a) Uniform distribution, (b) Low gain distribution, and, (c) Cluster distributions. For each configuration, we ran the experiments over 100 traces, each with 1000 jobs.

In Figure 3.4, we present the max-flow results for the various online algorithms for the case where users are uniformly distributed in a cell. The results are represented as a cumulative distribution of the max-flow obtained from each of the 100 traces. The y-axis represent the percentage of traces that had a max-flow that is less than x. The curve corresponding to *FIFO-continuous* reaches 100% first, which implies that the continuous algorithm has lower max-flow on the average than the other algorithms.

Practical Scenarios: Results

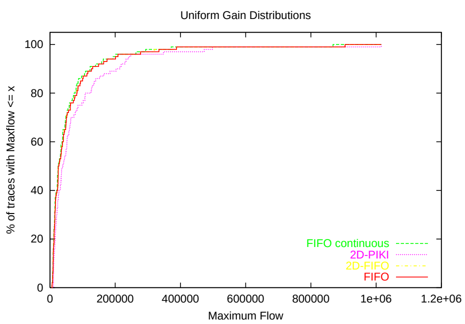


Figure 3.4: Uniform Distribution

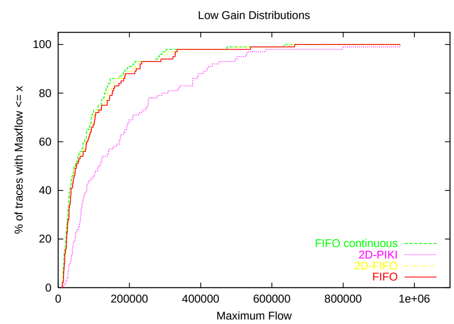


Figure 3.5: Low Gain Distribution

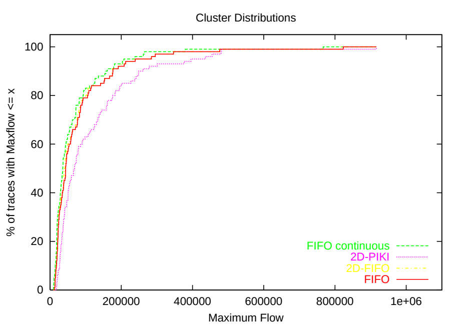


Figure 3.6: Cluster Distribution

In a low gain distribution, 75% of the users are near the edge of the cell, and hence, cannot achieve data rates more than 40 Kbps. The results for such a configuration are illustrated in Figure 3.5.

Finally, we arrange users in clusters around the cell, where some clusters are high gain clusters, and the others are low gain clusters. We show the results in this case in Figure 3.6.

Across all the traces and geographical distribution of users, the *2D-FIFO* algorithm is best in minimizing the maximum flow among the discrete-rate algorithms.

Resource Augmentation In this set of experiments, we will examine the amount of resource augmentation needed for a discrete-rate algorithm to achieve the same max-flow as *FIFO-continuous* and compare it to theoretical bounds given in Theorems 3.3.6 and 3.3.7.

We selected 50 web traces (at random) with 1000 jobs each, requested by 100 users, arriving over a period of 15 minutes. The users requesting the jobs were uniformly distributed over the region of the cell. The scheduling algorithms were provided with augmented power in steps $\{P, 1.25P, 1.5P, 1.75P, 2P\}$ and augmented number of codes in steps $\{C, 1.5C, 2C, 2.5C\}$. For each combination of augmented power and codes, we measured the max-flow at 100% of the demand. The demand was reduced in steps $\{s_i, 0.95s_i, 0.9s_i, 0.85s_i, 0.8s_i, 0.75s_i, 0.7s_i, 0.6s_i\}$ until the max-flow for the reduced demand in the discrete case was lesser than or equal to the max-flow in the continuous case with power P and codes C at 100% of the demand. We tested the *2D-FIFO* algorithm, since it outperforms the other algorithms in the average case.

The lower hull of the the reduced demand for each combination of augmented code and power was taken. This represents the maximum demand reduction for a given combination of augmented resources, as shown in Table 3.4. Two observations can be made here.

Augmentation factor		Demand
Power	Code	Reduction (in %)
1.0	1.0	85
1.0	2.5	85
1.25	1.0	90
1.25	2.5	90
1.5	1.0	100
2.0	2.0	100

Table 3.4: Resource augmentation: average case

(a) *The average case is vastly better than the worst-case.* We see that if power is augmented 1.5 times, then the max-flow in the discrete case equals that of the continuous case with power P. Thus, a system designed for meeting the QoS needs in the average case needs to over-provision its resources by an amount much lesser than that implied by the theoretical bounds.

(b) *Code augmentation and power augmentation are not the same.* As can be seen from the table, over-provisioning codes is not very efficient compared to over-provisioning of power. For example reducing the demand of 85% and using 2.5 times more codes is equivalent to using only 1.25 times more power and reducing the demand to only 90% of the original one.

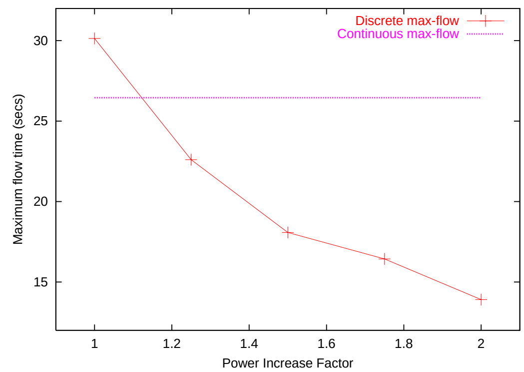


Figure 3.7: Max-flow with resource augmentation

Another interesting behavior is the variation is max-flow when the algorithm is provided with augmented resources, at 100% of demand. We illustrate this in Figure 3.7 for one of the web trace used earlier. The max-flow in the continuous case at 100% of demand without any augmentation is also shown for reference. This result reflects the asymmetric nature of code and power augmentation observed in the previous example. The max-flow did not change in the discrete case upon increasing the number of codes, even by a factor of 4. It was reduced for the continuous case, but the reduction was less than 0.05% when the number of codes increased 4 times. However, upon increasing power even by a small fraction, the max-flow decreased significantly, as is clear from the figure.

In conclusion, our experimental results demonstrate that the discrete-rate algorithms proposed in this section performs close to the optimal in the average case, even though their performance is theoretically unbounded. We also show that resource augmentation, in particular, power augmentation, will enhance the performance of discrete algorithms.

3.5 Related Work

There has been a significant amount of work on scheduling problems over wireless channels. We have studied the downlink scheduling problem. The uplink scheduling problem is a complementary problem where the fundamental issues are quite different. See [5] and references therein for more details.

Typically resource allocation problems study per-user rate throughput. The rate optimization problem has been extensively studied for various wireless system with focus varying from maximizing overall throughput to providing a minimum throughput guarantee for all users. A good discussion of related work about throughput optimization and fairness in wireless data networks can found in [57, 84].

Job scheduling is very popular in the context of processor scheduling, and various algorithms have been proposed for different QoS metrics such as completion time, maximum response time and, weighted average response time [28]. In wireless networks, job scheduling has been addressed in the context of downlink *broadcast* scheduling [76]. In a recent work, downlink unicast scheduling in CDMA systems was studied [44]; this is close to our work in spirit. However, they assume a linear rate model for the physical layer which is not accurate. Also, they do not have any upper bound on the number of available codes; hence, they study the problem of packing power only. We have studied the nuances of packing both power and code in this paper. Finally, we have provided a thorough competitive analysis of the online algorithms, in particular, using the resource augmented analysis; this is the first provable result known for any of the online scheduling problems, including the ones in [44].

3.6 Conclusions

In this chapter we have formulated new scheduling problems related to multiple rate, multiple code wireless networks. We focused our attention on the maximum response time criterion for packet scheduling. However, the formulation and approach can be extended to other criteria such as (weighted) average flow. More detailed results in this regard can be found in [51]. We proposed online algorithms that utilize the multicode, multirate feature of 3G/4G wireless networks by effectively assigning power and codes to different users and jobs. We performed experimental results to show that for several cases of practical interest the proposed algorithms perform much better than our worst case analysis shows. In summary, we have proposed simple online scheduling algorithms that effectively provide fine-grained QoS to the users by utilizing the advanced features of 3G/4G wireless networks.

Chapter 4

Bandwidth and Storage Allocation Problems under Real Time Constraints

The problem we study has been encountered in the context of the EU research project Euromednet on scheduling requests for remote medical consulting on a shared satellite UDP-TCP/IP channel [29]. Every request asks for a number of contiguous bandwidth slots to provide every request with a UDP-TCP/IP satellite connection between the users involved in the consulting. Bandwidth is assigned in slots of 64 kb/sec. The number of slots per end user depends on the type of service desired (typical values are 64 kb/sec for common internet services – 384 Kb/sec for audio/video.) At most 48 slots of 64 Kb/sec are available on the channel in this specific application. Requests also specify a duration of the consulting (typical values are from 1/2 hour to 2 hours), to be allocated within a time interval specified in the request. Requests, that are typically issued a few days in advance, are replied soon by the system with a positive or a negative answer on the basis of the pending requests and of the resources already allocated. Every accepted request is allocated starting from a base bandwidth for a contiguous number of slots along a time duration within the indicated time interval. The total bandwidth assigned to a single request must be contiguous due to the constraints imposed from FDMA (Frequency Division Multiple Access) technology. Other details regarding this specific application are available to [29].

The problem encountered in this application is a natural interesting combinatorial problem: every accepted request is scheduled on a rectangle in the time/bandwidth Cartesian space of basis equal to the duration and height equal to the requested bandwidth. Accepted requests must observe the *packing constraint* imposing no overlaps between any two scheduled rectangles. A benefit associated with every request indicates its relevance or the economic revenue gained from its acceptance. The objective is to maximize the overall benefit obtained from accepted requests. In the sequel of the chapter we denote this problem by Rectangle Packing (RP).

RP is related to a number of well studied combinatorial problems. Consider the machine scheduling problem with real time constraints in which every job asks to be scheduled without preemption for a given duration between a release time and a deadline. Only one job can be scheduled at any time on every single machine. A benefit is associated with every job with the goal of maximizing the benefit obtained from scheduled jobs. This is an old NP-hard scheduling problem [32]. Very recently the first constant approximation algorithms have been proposed [8] both on single and parallel machines.

A second related problem is the Dynamic Storage Allocation problem (DSA) where a set of requests for a contiguous area of memory along a specified time duration has to be allocated with the objective of minimizing the maximum storage space that is required. DSA is a classical problem in computer science [49] whose study backs to the sixties. The rectangle packing problem can be seen as a maximization version of DSA where storage space is limited and every process can be allocated within a prescribed time window rather than on a fixed interval. DSA has been shown to be tightly related to interval graph coloring. This relation has been exploited by Kierstead and Slusarek [46, 80] to give a 3-approximation algorithm for *aligned DSA*, i.e. the version of the problem where the storage space of every request is always a power of 2, that results in a 6 approximation for DSA. More recently, Gergov [34] proposed in a first paper a 5/2 approximation algorithm for aligned DSA, thus a 5 approximation for DSA, and in a second paper claimed a 3 approximation [35] for DSA.

A third closely related problem is the call control problem on a line network [31, 7] with capacities associated to every link. Requests are for establishing a connection between a pair of vertices at at given bandwidth and offer a given benefit if accepted. On every link it must be observed the *bandwidth constraint*, i.e. the overall bandwidth allocated on a link cannot exceed the capacity of the link. The objective is to maximize the benefit from accepted requests. In call control requests are assigned on a fixed interval of the line network and link may have different capacity, while in RP requests are allocated on one of a set of intervals within a time window while bandwidth availability is uniform along time. However, the major distinction between RP and call control is that call control only imposes the bandwidth constraint on every link, rather than the stronger packing constraint of RP.

We present a 12 approximation algorithm for the RP problem. As a basic step of the algorithm we solve a fractional LP problem for *aligned* RP, where bandwidth requests are power of 2, in which we only enforce the bandwidth constraint and requests can be fractionally accepted. We then show with a novel technique that the optimal fractional solution is a convex combination of a set of integral solutions, not yet feasible for RP, but holding a specific property called *stability*. We select the integral stable solution with highest benefit that can be partitioned into three feasible solutions of which we select the best one as the final solution of the algorithm. The approximation ratio we obtain is 6 for *aligned* RP and 12 in the general case. The proposed solution runs in pseudopolynomial time. It can be transformed into a strongly polynomial time algorithm at the expenses of a small increase in the approximation ratio. We also show a combinatorial algorithm with approximation ratio arbitrarily close to $26 + \epsilon$. This algorithm uses as a basic step the combinatorial algorithm devised in Bar-Noy et al. [6]. Independently from the results proposed in this dissertation, Bar-Noy et al. [6] proposed a 35 approximation for our problem that they call Benefit DSA. Their approach is to solve a version of the problem where requests are either accepted or rejected in an integral sense, while the packing constraint is relaxed to the milder bandwidth constraint. A solution of this problem is then combined with an algorithm for DSA. In a later version of their paper they improve the result to a $6\gamma - 1$ combinatorial approximation and to a $6\gamma - 3$ LP-based approximation, where γ is the approximation ratio for DSA. If we consider the 5-approximation for DSA of [34] this yields respectively a 29 combinatorial and a 27 LP-based approximation. The 3-approximation for DSA claimed in [35] yields a 17 combinatorial and a 15 LP-based approximation for RP.

We finally show how to extend our algorithm to the multiple channel case for bandwidth allocation or, equivalently, to the multiple storage devices case in the DSA problem. The rest of the chapter is organized as follows. In Section 4.1 we formally describe the RP problem. In Section 4.2 we describe the LP based approximation algorithm for the RP problem. In section 4.2.4 we show how the algorithm is turned into a strongly polynomial time algorithm. In Section

4.3 we present a combinatorial version of the algorithm. Finally, in Section 4.4 we describe the extension to multiple channels.

4.1 The RP problem

Given an input set of n requests $\{< r_i, d_i, b_i, l_i, \omega_i >\}_1^n$, where $r_i, d_i, b_i, l_i, \omega_i$ are integers, the generic request asks for a bandwidth interval of size b_i in $[0, B]$ along a time interval of length l_i in $[r_i, d_i]$. We will assume $b_i \leq 1$ and $B = 1$. A request can be either accepted or rejected. A request that is accepted is scheduled on a bandwidth interval $[f(i), f(i) + b_i]$ and a time interval $[t(i), t(i) + l_i]$ and offers benefit ω_i. An accepted request is represented with a rectangle of basis l_i and height b_i on a Cartesian space having the time on the abscissa and the bandwidth on the ordinate. A feasible schedule must observe the *packing constraint* imposing no overalap between any two rectangles. The objective of the algorithm is to maximise the overall profit obtained from accepted requests. The packing constraint will be sometime dropped in favour of the weaker *bandwidth constraint* merely imposing that the total bandwidth allocated at a time t does not exceed B. In the *aligned* version of RP every bandwidth request is a power of $1/2$.

4.2 A LP based approximation algorithm

We present an LP based approximation algorithm for LP. The algorithm is composed of three steps:

(1.) We first solve a fractional LP problem in which we only enforce the bandwidth constraint derived from an aligned version of RP by rounding all the bandwidth requests to the nearest higher power of 2.

(2.) We then show that the optimal solution to the fractional LP problem is a convex combination of a set of integral solutions not yet feasible but holding a property called *stability.* We select the best among these stable solutions with a benefit of at least $1/2$ the optimum to the LP problem.

(3.) In the final step of the algorithm we decompose the selected stable solution into three feasible solution of which we select that with highest benefit that will be the final output of the algorithm.

The obtained solution is a 6 approximation for aligned RP and a 12 approximation for the general problem.

4.2.1 The LP formulation

In this section we present the LP formulation we use as a basic step for the solution of the RP problem.

We first round every bandwidth request to the lowest higher power of $1/2$, namely $\overline{b}_i = \min_k \left\{ \frac{1}{2^k} : \frac{1}{2^k} \geq b_i \right\}$.
Variables x_{it}, $t = r_i, .., d_i - l_i$, are associated with request i. Variable x_{it} is ranging in $[0, 1]$, and denotes the schedule of request i with $t(i) = t$. We denote by x_{it} both a variable and its value in the LP relaxation. Constraint $\sum_{t=r_i}^{d_i-l_i} x_{it} \leq 1$ imposes that every request can be fractionally scheduled along a set of (possibly overlapping) intervals for an overall value at most one. Denote by T the latest deadline of a request and by $\mathcal{T} = \{1, 2, \ldots, T\}$. We also write the bandwidth constraint at any time $t \in \mathcal{T}$, namely that the overall bandwidth assigned to the requests fractionally scheduled at time t is at most 1.

$$
\begin{aligned}
\max \sum_{i=1}^{n} \sum_{t=r_i}^{d_i-l_i} \omega_i x_{it} & \\
\sum_{i,t' : t \in [t', t'+l_i)} \overline{b}_i x_{it'} \leq 1, & \quad \forall \quad t \\
\sum_{t=r_i}^{d_i-l_i} x_{it} \leq 1, & \quad \forall \quad i \\
x_{it} \in [0, 1], & \quad \forall \quad t, i \\
x_{it} = 0, & \quad \forall \quad i, t \notin [r_i, d_i - l_i]
\end{aligned}
$$

Lemma 4.2.1 *For any instance of RP it holds $OPT(LP) \geq OPT(RP)/2$. For any instance of aligned RP it holds $OPT(LP) \geq OPT(RP)$.*

Proof: Consider a new formulation LP_1 obtained from LP by replacing variables $\overline{b}_i$ with the original b_i in the bandwidth constraint, namely

$$\sum_{i,t' : t \in [t', t'+l_i)} b_i x_{it'} \leq 1, \forall t, \tag{4.1}$$

and by imposing the integrality constraints $x_{it} \in \{0,1\}$. Since $\overline{b}_i \le 2b_i$, if a set of x_{it} is a solution to LP_1 then $\frac{x_{it}}{2}$ is a solution to LP with benefit at least $\frac{1}{2}$ the benefit of LP_1, for which $OPT(LP) \ge \frac{1}{2}OPT(LP_1)$. We also observe that any solution to RP is a solution to LP_1, for which $OPT(LP_1) \ge OPT(RP)$. Then $OPT(LP) \ge OPT(LP_1)/2 \ge OPT(RP)/2$.

For the aligned case, we simply obtain $OPT(LP) \ge OPT(RP)$. ■

4.2.2 The algorithm for obtaining a stable solution

We present the algorithm for finding a stable integral solution starting from a fractional solution to the LP problem. We denote by i^t the request i scheduled at time t and we say that i^t is a copy of request i.

Definition 4.2.1 *Given a schedule of requests, the support at time t', denoted by support(t'), is the maximum value such that there exists a set of j non-overlapping requests $i_1, i_2, .., i_j$ scheduled at time t' for which $f(i_1) = 0$, $f(i_k) = f(i_{k-1}) + b_{i_{k-1}}$, $k = 2, .., j$, $f(i_j) =$ support(t').*

Request i^t is (h, t') stable if $h =$ support$(t') = max_{t'' \in [t, t+l_i)}$support$(t'')$.

A schedule of requests is stable if every request i in the schedule is (h_i, t_i) stable for some h_i and t_i.

The geometrical interpretation of a request i (h, t) stable is a rectangle placed on the top of a pile of non-overlapping rectangles of total bandwidth h (see Figure 4.1). We will say that the rectangles in the pile form the support of i. Observe that 2 requests in a stable solution can overlap.

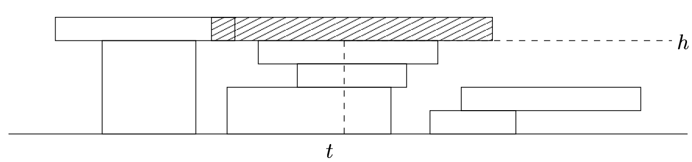


Figure 4.1: The rectangle associated with request i is filled in the figure. Request i is (h, t) stable.

Given a solution x_{it} to the LP problem we denote by α the largest value such that every x_{it} is an integer multiple of α.

Algorithm Stable selects a stable solution by constructing at most $2/\alpha$ integral stable solutions and then choosing that one with highest benefit. Denote by $\mathcal{S}$ the set of solutions constructed by the algorithm and by s the cardinality of $\mathcal{S}$ at a generic step of the algorithm.

Algorithm Stable is composed of the following steps:

Algorithm Stable:

Input: the fractional optimal solution to LP

1. Order the non-zero x_{it} by non increasing $\overline{b}_i$.

2. Consider $\frac{x_{it}}{\alpha}$ copies for every request i^t.

3. $\mathcal{S} = \emptyset$

4. Assign every copy of i^t to a solution as follows:

   (a) Select those solutions $S_1, ..., S_m$, out of the s solutions constructed until now, not containing a copy of request i, .

   (b) Merge the m solution $S_1, ..., S_m$ into a single solution $S(m)$ of bandwidth m.(The relative order of the solution is not relevant for the algorithm.)

   (c) Let the replication of i^t be (h, t') stable in $S(m)$.

   (d) If $h < m$, then assign the copy of i^t to solution $S_{\lfloor h \rfloor + 1}$ with $f(i^t) = h \bmod 1$; If $h = m$, then construct a new solution having i^t assigned with $f(i^t) = 0$ and add it to $\mathcal{S}$.

5. Select the solution in $\mathcal{S}$ with highest benefit that we call S_{best}.

Every solution S constructed by the algorithm is clearly stable and verifies the property that for each request i S contains at most one copy i^t. We now show that for every copy of i^t, $f(i^t) + \overline{b}_i \leq 1$.

We first give a preliminary Lemma.

Lemma 4.2.2 *For every request i and every solution S, $f(i) = k\overline{b}_i$ for some integer k.*

Proof: The rectangles in the support of i are ordered by non increasing bandwidth. Since the height of each rectangle is a power of $1/2$, we have that h_i is a multiple integer of $\overline{b}_i$. ■

Lemma 4.2.3 *For every copy of a request i^t, $f(i^t) + \overline{b}_i \leq 1$.*

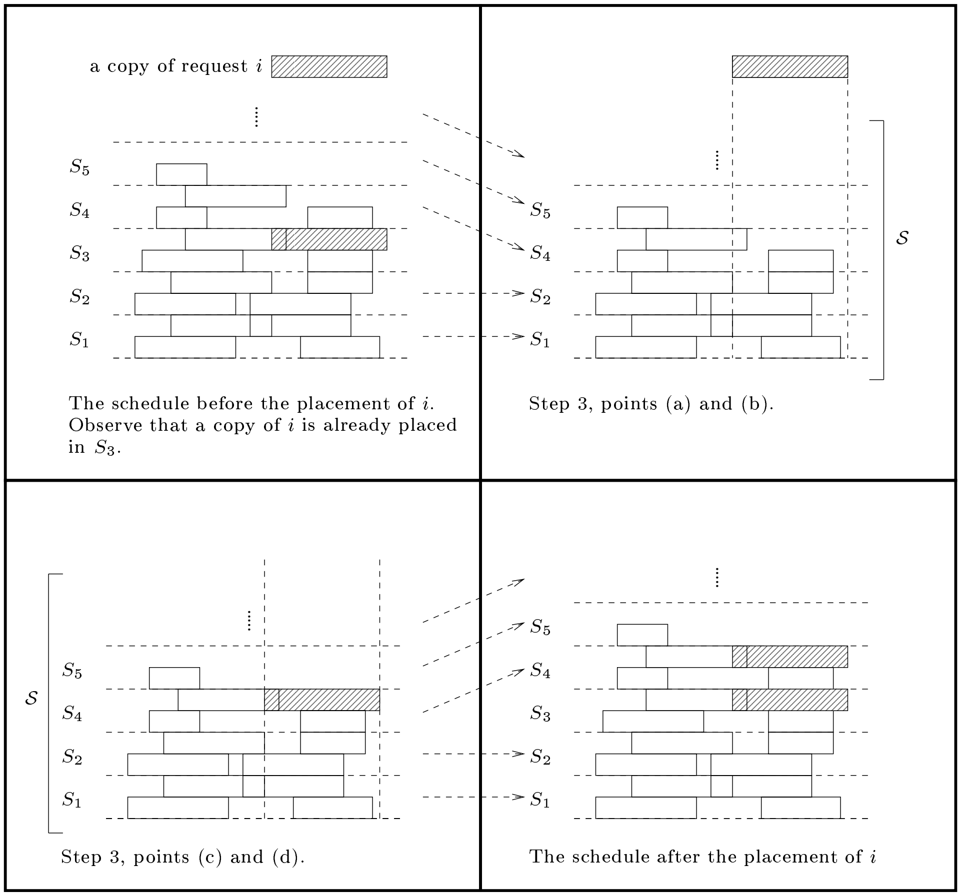


Figure 4.2: The algorithm for obtaining a stable solution.

Proof: By the definition of the algorithm every copy of a i^t is scheduled at $f(i^t)$ if $(f(i^t), t')$ stable with $f(i^t) < 1$. By the previous Lemma we have that $1 - f(i^t)$ is a multiple integer of $\overline{b}_i$ for which the thesis follows. ■

The following Lemma bounds the number of solutions in $\mathcal{S}$.

Lemma 4.2.4 *The number of solutions in $\mathcal{S}$ at the end of algorithm Stable is at most $s = \frac{2}{\alpha}$.*

Proof: We prove the claim by showing that every i^t is (h, t') stable for a value $h < m$. Assume by contradiction $h \geq m$. At most $1/\alpha$ distinct copies of i^t are allocated for every request i. Since $2/\alpha$ solutions are available, at least $m \geq 1/\alpha + 1$ solutions $S_1, .., S_m$ not containing a copy of i are available for a single i^t. If $h \geq m$ then at some time t' the whole bandwidth has been assigned for the whole m solutions, namely for any S_j, $\sum_{i^t \in S_j : t' \in [t, t+l_i)} \overline{b}_i = 1$.

From the packing constraint in the LP problem, we have that at time t': $\alpha \sum_{i,t:t' \in [t,t+l_i)} \frac{\overline{b}_i x_{it}}{\alpha} \leq 1$.

It follows that at time t':

$$1 \geq \alpha \sum_{i,t:t' \in [t,t+l_i)} \frac{x_{it}\overline{b}_i}{\alpha} = \alpha \sum_{S_j} \sum_{i^t \in S_j : t' \in [t,t+l_i)} \overline{b_i} \geq m\alpha \geq 1 + \alpha,$$

thus a contradiction. ■

Observe that the simpler alternative would just place every i^t in the first solution where it fits, i.e. where i^t is (h,t) stable with $h \leq 1 - \overline{b}_i$, if any. However this alternative fails to locate all the replications into at most $2/\alpha$ solutions.

Lemma 4.2.5 *Algorithm* Stable *runs in time polynomial in n and $1/\alpha$.*

Proof: First observe that the overall number of copies of i^t's associated with non-zero x_{it}'s is polynomial in n and $1/\alpha$ and that the running time of the algorithm is dominated by step 4.

Steps 4(a) (b) and (d) can be easily implemented to run in time polynomial in n and $1/\alpha$. We are left to show that step 4(d) can be implemented in time polynomial in n and $1/\alpha$, i.e. we can find in polynomial time for a copy i^t values h and t' such that i^t is (h,t') stable in the merge of m solutions $S_1, S_2, \ldots, S_m$ denoted in the following by $\mathcal{S}'$. Let $\mathcal{T}'$ be the set of time instants such that every $t' \in \mathcal{T}'$ is either the starting or the ending time instant of a request scheduled in $\mathcal{S}'$. The number of time instants in $\mathcal{T}'$ is bounded by the total number of copies that are scheduled over all the solutions, and therefore polynomial in n and $1/\alpha$. Request i^t is certainly (h,t') stable for a time instant $t' \in \mathcal{T}'$. Given a time t', to determine if i^t is (h,t') stable can also be done in polynomial time for which the claim of the lemma follows. ■

4.2.3 Constructing a feasible solution

In this section we show how the stable solution S_{best} can be decomposed into three feasible solutions to the *RP* problem. The best among the three solutions has beenfit at least 1/3 the benefit of S_{best}.

We construct the intersection graph of S_{best} by assigning a vertex to every rectangle and connecting with an edge every pair of vertices representing intersecting rectangles. The resulting graph is shown to be 3-colourable in polynomial time. (This holds not only for S_{best} but for any stable solution.) Every of the three sets of rectangles of same color forms a feasible solution since the rectangles are non overlapping and every request appears at most once.

The algorithm is as follows:

1. Construct the intersection graph of S_{best};
2. Colour the intersection graph with three colours with the following algorithm:
   (i.) Consider rectangles in order of non increasing bandwidth $\overline{b}_i$;
   (ii.) Colour rectangles with same $\overline{b}_i$ and $f(i)$ in order of increasing starting point, every time using one of the 3 colours not assigned to the intersecting rectangles.
3. Accept those rectangles assigned with same colour with highest total benefit;
4. Reduce every rectangle's height $\overline{b}_i$ to the original b_i.

We need some properties of a stable schedule to prove that the algorithm gives a legal 3-coloring of the graph. The following is a corollary of Lemma 4.2.2.

Corollary 4.2.6 *Consider two requests i and j in S_{best} with $\overline{b}_i \geq \overline{b}_j$, and assume they are (h_i, t_i) and (h_j, t_j) stable. Request i intersects with request j only if $h_i \leq h_j < h_i + \overline{b}_i$.*

Lemma 4.2.7 *Consider a schedule with two intersecting requests i and j that are respectively (h_i, t_i) and (h_j, t_j) stable. It holds $t_i, t_j \notin [t(i), t(i) + l_i) \cap [t(j), t(j) + l_j)$.*

Proof: The proof is by contradiction. Assume request i placed before j and hence $\overline{b}_i \geq \overline{b}_j$. If $t_i \in [t(i), t(i)+l_i) \cap [t(j), t(j)+l_j)$ then i is part of the support of j, $h_j \geq h_i + \overline{b}_i$, a contradiction since the two rectangles are overlapping.

Assume $t_j \in [t(i), t(i) + l_i) \cap [t(j), t(j) + l_j)$. By Corollary 4.2.6 it must be $h_j \geq h_i$. Since we are considering the aligned case, at least one rectangle of the support of j in t_j, say h, will be scheduled between $h_i - \overline{b}_h$ and h_i. Therefore, i is part of the support of j, a contradiction. ■

The next Lemma states that if the rectangles i and j are intersecting the two associated time intervals are not nested.

Lemma 4.2.8 *For any two intersecting requests i, j, it never holds $[t(i), t(i)+l_i) \subseteq [t(j), t(j)+l_j)$.*

Proof: The proof is by contradiction. If i and j are overlapping and $[t(i), t(i)+l_i) \subseteq [t(j), t(j)+l_j)$ then for the support of i it holds $t_i \in [t(i), t(i)+l_i) \cap [t(j), t(j)+l_j)$, a contradiction to Lemma 4.2.7. ∎

Lemma 4.2.9 *The maximum clique size of the intersection graph is 2.*

Proof: Assume by contradiction that requests i, j and k form a clique of size 3 and that k is placed in S_{best} after i and j. Assume i is (h_i, t_i) stable, j is (h_j, t_j) stable and $t_i \leq t_j$. Request k must be completely contained in the interval (t_i, t_j), otherwise k is either $(h_i + \overline{b}_i, t_i)$ stable or $(h_j + \overline{b_j}, t_j)$ stable, thus it does not intersect with i or j.

Therefore $[t(k), t(k) + l_k)$ is completely contained in (t_i, t_j) leading to the fact that either $t_k \in (t(i), t(i) + l_i) \cap (t(k), t(k) + l_k)$ or $t_k \in (t(j), t(j) + l_j) \cap (t(k) + l_k)$. By Lemma 4.2.7 this is a contradiction to the assumption that k intersects both i and j. ∎

We finally prove that the algorithm produces a legal 3 colouring of the intersection graph.

Theorem 4.2.10 *The algorithm colours the intersection graph with 3 colours.*

Proof: By Corollary 4.2.6, requests with same $\overline{b}_i$ and different $f(i)$ are non intersecting. Therefore they can be coloured independently. Concentrate on a set of requests with same $\overline{b}_i$ and $f(i)$. They are coloured greedily in order of starting point, i.e. from left to right.

Consider one such request i. By Lemma 4.2.8, every request intersecting i can intersect either $t(i)$ or $t(i) + l_i$, but not both endpoints. If more than one request intersects an endpoint of i, by Corollary 4.2.6, these all intersect in that point thus creating a clique of size at least 3, by Lemma 4.2.9 a contradiction. Therefore at most 1 request intersects each endpoint of i, at most 2 requests intersects i, therefore always leaving one colour available for i. ∎

We finally show the approximation ratio we obtain.

Theorem 4.2.11 *There exists an algorithm for the RP problem that is 12-approximated in the general case and 6-approximated in the aligned case.*

Proof: The algorithm selects a solution S_{best} whose benefit is at least $OPT(LP)/2$ as it follows from:

$$OPT(LP) = \sum_{S_l} \sum_{i^t \in S_l} \alpha\omega_i \leq \frac{2}{\alpha} \sum_{i^t \in S_{Best}} \alpha\omega_i \leq 2 \sum_{i^t \in S_{Best}} \omega_i.$$

By Lemma 4.2.1 $OPT(RP) \geq OPT(LP)/2$ in the general case for which the benefit of S_{best} is at least 1/4 the benefit $OPT(RP)$, while in the aligned case we have $OPT(RP) \geq OPT(LP)/2$. Moreover we colour the requests of S_{best} with 3 colours and select the set of intervals of highest benefit with same colour, for which the final solution has benefit at least 1/3 the benefit of S_{best}. Altogether we obtain an approximation ratio of 12 for the general case and of 6 for the aligned case. ■

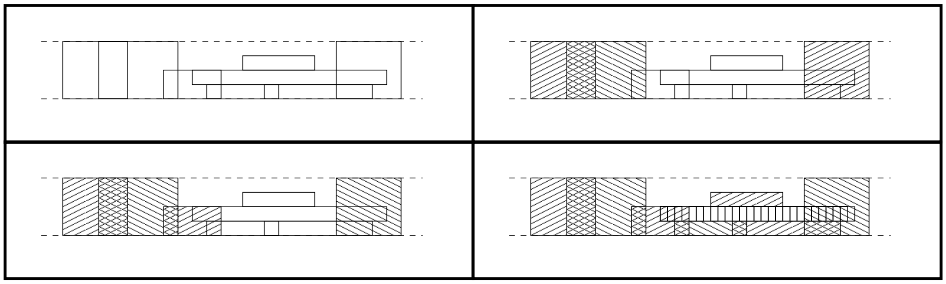


Figure 4.3: Requests are coloured by non-increasing bandwidth size.

4.2.4 A strongly polynomial time algorithm

The running time of the previous algorithm for the RP problem might not be polynomial for two reasons:

1. the number of constraints in the linear programming formulation LP depends on the number of time slots T and might be exponential in the input size;

2. The maximum number $2/\alpha$ of stable solutions and the maximum number of copies $1/\alpha$ of a request assigned to stable solutions might not be polynomial in the input size.

We propose a strongly polynomial time algorithm based on a technique used in [8]. The polynomial algorithm is composed of the following steps:

1. Solve a new LP' formulation with a polynomial number of constraints; let X' be the solution to LP';

2. Apply procedure Stable to a solution X'' obtained from X' by rounding down every variable assignment to to the closest multiple of $1/n^4$;

3. Construct a feasible solution as shown in the previous section.

Since all release times, deadlines and processing times are integral we assume that each request is processed at an integral point of time. In the LP' formulation they are furtherly restricted to a set of integers polynomial in n, thus obtaining a number of variables and constraints of the linear programming formulation that is polynomially bounded.

The modified linear programming formulation LP' defines *big slack* requests and *small slack* requests. A request i is *big slack* if $d_i - r_i > n^2 \times l_i$, *small slack* otherwise.

A big slack request i is completely scheduled up to 1 in LP' for a fraction $1/n^2$ on each of n^2 disjoint time intervals between r_i and d_i. The bandwidth allocated to big slack request at any time is then bounded by $1/n$.

A small slack request i may be scheduled at one of $n^2 + 1$ time divisors between r_i and d_i. The distance between two divisors is less than l_i. The set $\mathcal{T}'$, $\mathcal{T}' \subseteq \mathcal{T}$ in the modified LP' is the union of all time divisors for the n requests. $\mathcal{T}'$ has cardinality $O(n^3)$.

Let $OPT(LP')$ be the optimal value of the LP' relaxation; since at each instant the overall bandwidth allocated for big slack requests is less than n^{-1} it follows that $OPT(LP) \leq (1 + 2n^{-1})OPT(LP')$.

Note that the bandwidth constraint may be satisfied in LP' at two successive time divisors t_1 and t_2 but violated in the middle due to the overlap between a request ending at t_1 and a request starting at t_2. The first one may actually end between t_1 and t_2 while the second one may start between t_1 and t_2. This will lead the procedure stable to construct $3/\alpha$ stable solutions to place all the copies of the requests, rather than $2/\alpha$ as shown in the previous section.

Remind that procedure Stable is applied to a solution X'' obtained by rounding down the optimal solution to LP'. We will apply the following lemmas to solution X'' obtained by rounding down the optimal solution to LP' is a fesible solution to LP'.

Lemma 4.2.12 *Given a feasible solution to LP' then algorithm* Stable *constructs at most $\frac{3}{\alpha}$ stable solutions.*

Proof: The proof is similar to the proof of Lemma 4.2.4. At most $1/\alpha$ distinct copies of i^t need to be placed for every request i. Since $3/\alpha$ solutions are available, at least $m \geq 2/\alpha + 1$ solutions $S_1, .., S_m$ not containing a copy of i are available for a single replication of i^t. By contradiction, if no stable solution can host the new request, then every solution has allocated at any point a bandwidth that is at least $\alpha/2$, for a total over all the solutions bigger than 1. ∎

Theorem 4.2.13 *There exists a strongly polynomial algorithm for the RP problem that is* $18 + O(n^{-1})$ *approximated in the general case and* $9 + O(n^{-1})$ *approximated in the aligned case.*

Proof: For the running time observe that both the number of constraints in the linear programming formulation LP' and the value of α are polynomial in n; by Lemma 4.2.5 the running time of the whole algorithm is also polynomial in n.

The proof of the approximation ratio is similar to the proof of Theorem 4.2.11. Let $OPT(LP)$ and $OPT(LP')$ be the values of the optimal solutions to LP and LP' and $R(LP')$ be the value of the solution X'' obtained by rounding down to the nearest multiple of $1/n^4$ every variable assignment in solution X' to LP'. Let ω_{max} be the maximum benefit obtainable by accepting a single request; clearly $OPT(LP)$ is at least ω_{max}.

Since the number of variables in LP' is at most $n^3 + n$ and each nonzero variable of X' is rounded to the closest multiple of n^{-4} it follows that $OPT(LP') - R(LP') \leq \omega_{max}(n^{-1} + n^{-3})$ and, therefore, for sufficiently large n,

$$OPT(LP') \leq R(LP')(1 + 2n^{-1}).$$

Since $OPT(LP) \leq (1 + 2n^{-1})OPT(LP')$ we also have for a large enough n, $OPT(LP) \leq R(LP')(1 + 5n^{-1})$.

Lemma 4.2.12 implies that the benefit of the solution S_{best} obtained by the procedure Stable is at least $R(LP')/3$ and Lemma 4.2.1 implies that $OPT(RP) \leq 2OPT(LP)$ in the general case and $OPT(RP) \leq OPT(LP)$ in the aligned case. Since we colour the requests of S_{best} with 3 colours and select the set of intervals with same colour of highest benefit we obtain an approximation ratio of $18 + O(n^{-1})$ for the general case and of $9 + O(n^{-1})$ for the aligned case.
■

4.3 A combinatorial algorithm

In this section we sketch how to replace the basic step of the approximation algorithm based on the solution of a fractional LP formulation with a combinatorial algorithm that delivers a constant approximation solution to the LP problem.

We partition the requests into *wide* requests, that ask at least 1/2 of the available bandwidth, and *narrow* requests whose bandwidth requirement is less than 1/2. We solve the RP problem separately for wide requests and narrow requests and we choose the best solution. If all requests are wide then RP is equivalent to interval scheduling for which a 2 approximation algorithm is known [81].

For narrow requests we replace the basic step of the algorithm based on solving the LP formulation with a combinatorial algorithm. We divide every request in k identical requests each one with a fraction $1/k$ of the bandwidth and of the profit of the original request. We then apply the combinatorial algorithm of [6] for finding an approximate integral solution to the problem in which the only bandwidth constraint is imposed. Lemma 3.2 of [6] states the following:

Lemma 4.3.1 *For each integer k there exists a combinatorial algorithm that finds a $2 + 1/k$ approximate solution to the LP formulation if all requests are narrow.*

Recall that the algorithm of section 3 can be applied to any feasible solution of the linear programming relaxation. Therefore the combinatorial algorithm gives a solution that is away from the optimal LP solution for at most a $2+\frac{1}{k}$ factor thus leading to a $12(2+\frac{1}{k})$ approximate solution for narrow requests. Combined with the 2 approximation for wide requests we obtain:

Theorem 4.3.2 *For every k there exists a $26+1/k$ combinatorial approximation algorithm for the RP problem.*

Proof: Given k and an instance of the RP problem, the algorithm consider two problems; the first one is obtained by considering only wide requests and the second one by considering only narrow requests. We denote with $OPT(RP_w)$ and S_w the values of the optimal and of a 2 approximate solution on wide requests; analougously we denote with $OPT(RP_n)$ and S_n the values of the optimal and of a $(24 + 1/k)$ approximate solution for narrow requests. $A(RP)$ denotes the value of the solution found by the algorithm that is at least $\max(S_w, S_n)$.

Clearly the optimal solution $OPT(RP)$ of the given instance satisfies

$$OPT(RP) \leq OPT(RP_w) + OPT(RP_n) \leq 2S_w + (24 + \frac{1}{k})S_n.$$

We distinguish two cases: if $S_w \leq S_n$ then

$$OPT(RP) \leq 2S_w + (24 + \frac{1}{k})S_n \leq (26 + \frac{1}{k})S_n \leq (26 + \frac{1}{k})A(RP).$$

If $S_w > S_n$ then

$$OPT(RP) \leq 2S_w + (24 + \frac{1}{k})S_n \leq (26 + \frac{1}{k})S_w \leq (26 + \frac{1}{k})A(RP).$$

■

4.4 The multiple channel case

In this section we assume that m channels, each one with a bandwidth $B_j \leq 1$, are available. For the sake of simplicity we assume the B_j's to be powers of 1/2. We briefly sketch the extension of known techniques [8], to obtain a $c+1$ throughput maximization approximation algorithm for m parallel unrelated machines provided a c algorithm for a single machine. We consider a Linear Programming formulation with variables x_{ijt} indicating the allocation of request i at time t on machine j. We set $x_{ijt} = 0$ for those machines j where $\overline{b}_i > B_j$. We then solve the LP problem and apply our rounding algorithm in order from channel 1 to channel m while we disregard on channel j requests already accepted on a previous channel. The analysis shown in [8] allows to conclude with the following theorem:

Theorem 4.4.1 *Provided a c approximation algorithm for the RP problem on a single channel, there exists a $c+1$ approximation algorithm for the RP problem on multiple channels.*

4.5 Conclusions

In this chapter we have presented constant approximation algorithms for the RP problem, a throughput version of bandwidth and storage allocation problems when real time constraints are imposed. Our algorithm uses as a basis a solution of a Linear Programming formulation and partitions it into a convex combination of integral solutions with a novel rounding technique. We improve the approximation results found independently from our work in [6].

An interesting open problem is to study the problem in the on-line model in which requests for bandwidth allocation are presented over time. In the on-line setting we also observe that in reality rejection of the requests may not be allowed if enough bandwidth is available. We finally mention the improvement of the approximability of the problem, in particular by exploiting some of the ideas behind the recent work by Gergov on approximating DSA [34, 35].

Chapter 5

Randomized Lower Bounds for Online Path Coloring

In this chapter we present randomized lower bounds for on-line path coloring problems on line and tree networks, a special class of on-line graph coloring problems. The input instance to an online graph coloring problem is a sequence $\sigma = \{v_1, ..., v_{|\sigma|}\}$ of vertices of a graph. The algorithm must color the vertices of the graph following the order of the sequence. When the color is assigned to vertex v_i, the algorithm can only see the graph induced by vertices $\{v_1, ..., v_i\}$. The goal of a graph coloring algorithm is to use as few colors as possible under the constraint that adjacent vertices receive different colors.

Online graph coloring problems have been studied by several authors [48, 39, 55, 85]. The study of online graph coloring has actually been started even before the notion of competitive analysis of online algorithms was introduced [79]. Kierstead and Trotter [47] in 1981 considered the online coloring problem for interval graphs. Every vertex of an interval graph is associated with an interval of the line. Two vertices are adjacent if the two corresponding paths s are edge-intersecting. Since interval graphs are perfect graphs [37], they have chromatic number χ equal to the maximum clique size ω, i.e. the maximum number of intervals overlapping at a single of the line.

In [47] a deterministic online algorithm that colors an interval graph of chromatic number ω with $3\omega - 2$ colors is presented. They also prove that the $3\omega - 2$ bound is tight: for every deterministic algorithm there exists an input sequence where the algorithm uses at least $3\omega - 2$ colors.

The problem of coloring online an interval graph abstracts a set of scheduling problems in wireless communication. Consider a set of requests that arrive over time. Each request specifies a contiguous time interval in which it has to be served. According to the FDMA technology, two overlapping requests must use different frequencies, namely two overlapping intervals (vertices of an interval graph) must be colored with different colors. The problem is on-line since the assignment of frequencies to requests must be done upon arrival even if the requests have to be served in the future.

The online coloring of paths on a tree network naturally gives rise to an on-line graph coloring problem. We assume the tree network known in advance to the algorithm. Every path on the tree network presented in the sequence is associated with a vertex of the graph. Two vertices are adjacent in the graph if the two corresponding paths are intersecting. The graph that is formed is called the *intersection graph*. This problem has recently received a growing attention due to its application to wavelength assignment in optical networks [70, 10, 36].

An $O(\log n)$ competitive deterministic algorithm for the problem of on-line coloring paths on a tree network has been shown by several authors (see for instance [10, 36]). Bartal and Leonardi [10] also show an almost matching $\Omega(\Delta/\log\Delta)$ deterministic lower bound on a tree of diameter $\Delta = O(\log n)$, where n is the number of vertices of the graph.

In this dissertation we present the first lower bounds on the competitive ratio of *randomized* algorithms for online interval graph coloring and online coloring of paths on tree networks.

Randomized algorithms for online problems [11] have often been proven to achieve competitive ratios that are strictly better than deterministic online algorithms. The competitive ratio of a randomized algorithm against an oblivious adversary is defined as the maximum over all the input sequences of the ratio between the expected online cost and the optimal offline cost. The input sequence for a given algorithm is generated by the oblivious adversary without knowledge of the random choices of the algorithm. However, there is no known randomized on-line coloring algorithm for any network topology that achieves a competitive ratio that is substantially better than the best deterministic algorithm for the problem.

In this thesis we present the first randomized lower bound, up to our best knowledge, for online coloring of interval graphs. We show that any randomized algorithm uses an expected number of colors equal to $3\omega - 2 - o(1/\omega)$ for an interval graph of maximum clique size equal to ω, thus proving that randomization does not allow to substantially improve upon the best deterministic algorithm of [47], answering an open question posed in [19].

Our second result is a first randomized $\Omega(\log \Delta)$ lower bound for online coloring of paths on a tree network of diameter $\Delta = O(\log n)$. There is still a substantial gap between the presented lower bound and the $O(\log n)$ deterministic upper bound known for the problem.

The current status of the online path coloring problem on trees can be compared with the known results for the dual problem of selecting online a maximum number of edge-disjoint paths on a tree network, i.e. a maximum independent set in the corresponding intersection graph. An $O(\log \Delta)$-competitive randomized algorithm is possible for the online edge-disjoint path problem on trees [2, 53], that compares with a matching $\Omega(\Delta)$ deterministic lower bound obtained on a line network of diameter $\Delta = n$ [1]. Our result still leaves open the question if an $O(\log \Delta)$-competitive randomized algorithm is possible for the online path coloring problem on tree networks.

Our result has also implications on on-line coloring of inductive graphs. A graph is d-inductive if the vertices of the graph can be associated with numbers 1 through n in a way that each vertex is connected to at most d vertices with higher numbers. Irani [40] shows that any d-inductive graph can be colored online with $O(d \log n)$ colors and presents a matching $\Omega(\log n)$ deterministic lower bound. The graph obtained from the intersection of paths on a tree network has been independently observed to be a $(2\omega - 1)$ inductive graph by [10] and by Kleinberg and Molloy as reported in [19]. Our lower bound for online path coloring on trees then implies a first $\Omega(\log \log n)$ lower bound on the competitive ratio of randomized algorithms for online coloring of inductive graphs.

We conclude this section by mentioning the previous work on randomized online coloring algorithms for general graphs. Vishwanathan [85] gives an $O(n/\log n)$ competitive randomized algorithm, improving over the $O(n/\log^* n)$ deterministic bound of Lovász, Saks and Trotter [55]. In [85] it is also presented an $\Omega((1/(\chi-1)((\log n/(12(\chi+1)))-1)^{\chi-1})$ randomized lower bound for coloring on-line a graph of chromatic number χ. However, such result is obtained for the model in which the algorithm does not know in advance a the graph from which a subgraph is presented. Halldórson and Szegedy [39] give an $\Omega(n/\log^2 n)$ randomized lower bound for the problem. Bartal Fiat and Leonardi [9]first study the model in which the graph G from which the vertices to color are drawn is known in advance to the online algorithm. The sequence σ may contain only a subset of the vertices of G. The algorithm must color the subgraph of G induced by the vertices of σ. The authors show that even under this model an $\Omega(n^{\epsilon})$ randomized lower bound, for a fixed $\epsilon > 0$, is possible.

The chapter is structured as follows. Section 5.1 presents the lower bound on online coloring of interval graphs. Section 5.2 presents the lower bound for path coloring on tree networks. Conclusions and open problems are in Section 5.3.

5.1 A lower bound for online interval graph coloring

In this section we present a lower bound on the competitive ratio of randomized algorithms for online interval graph coloring.

The input instance to the online interval graph coloring problem is given by a sequence of intervals on a line graph. Every interval is denoted by two endpoints of the line. The algorithm must color the intervals one by one, in the order in which they appear in the sequence. The goal is to use as few colors as possible under the constraint that any two overlapping intervals are assigned different colors.

The competitive ratio of an online algorithm for the interval graph coloring problem is given by the maximum over all the input sequences of the ratio between the expected number of colors used by the algorithm and the chromatic number of the interval graph, i.e. the maximum number ω of intervals overlapping at a single point of the line.

A lower bound for randomized algorithms against an oblivious adversary is established using the application of Yao's Lemma [94] to online algorithms [19, 9]. A lower bound over the competitive ratio of randomized algorithms is obtained by proving a lower bound on the competitive ratio of deterministic online algorithms on a specific probability distribution over the input sequences for the problem.

We first give some notation. We will denote by $P_{\overline{m}}$ the specific probability distribution over input sequences of chromatic number ω we use to prove the lower bound. Probability distribution $P_{\overline{m}}$ will be described by a set of input sequences with chromatic number $\overline{m}$, *with every input sequence presented with equal probability.*

We denote by $\sigma \in P$ the generic input sequence of probability distribution P. We slightly abuse notation by denoting with σ also the set of intervals $\{I_1, ..., I_{|\sigma|}\}$ in the sequence.

Probability distributions P and Q are said *independent* if for any $I^1 \in \sigma^1 \in P$, $I^2 \in \sigma^2 \in Q$, I^1 and I^2 are disjoint intervals. The set of sequences of probability distribution $P \cup Q$ is obtained by the concatenation of every input sequence of P with every input sequence of Q.

5.1.1 The probability distribution

The probability distribution P_ω used for proving the lower bound is recursively defined. We will resort to a pictorial help to describe the sequence.

Probability distribution P_1 is formed by a single input sequence containing a single interval. P_ω is the union of λ *independent* and *identical* probability distributions $\overline{P}^1_\omega, ..., \overline{P}^\lambda_\omega$, as described in Figure 5.1. The value of λ will be fixed later.

$\overline{P}^j_\omega$ is obtained from four independent distributions $P^1_{\omega-1}, P^2_{\omega-1}, P^3_{\omega-1}, P^4_{\omega-1}$. The set of input sequences of $\overline{P}^j_\omega$ is obtained by the concatenation of every input sequence of $P^1_{\omega-1} \cup P^2_{\omega-1} \cup P^3_{\omega-1} \cup P^4_{\omega-1}$ with every of the 10 distinct subsequences $T_1, ..., T_{10}$, called *configurations,* of at most 4 intervals described in Figure 5.2. The intervals of every of the 10 different configurations are numbered in Figure 5.2 following the order in which they appear in the sequence. Every probability distribution $P^i_{\omega-1}$ is generated by applying the present definition for $\omega - 1$.

Observe that every input sequence of $P_{\overline{m}}$ has chromatic number $\overline{m}$. This can easily be seen with an inductive argument. Probability distribution P_1 contains a single sequence with chromatic number 1. By induction, every input sequence from $P^i_{\omega-1}$, $i = 1, .., 4$, has chromatic number $\omega - 1$. Every input sequence $\sigma \in P^1_{\omega-1} \cup P^2_{\omega-1} \cup P^3_{\omega-1} \cup P^4_{\omega-1}$ has also chromatic number $\overline{m} - 1$. One can check from Figure 5.2 that the concatenation of σ with every of the 10 configurations increases the clique size and then the chromatic number by exactly 1. Since $P_{\overline{m}}$ is the union of λ independent probability distributions $\overline{P}^1_\omega, ..., \overline{P}^\lambda_\omega$, every sequence of $P_{\overline{m}}$ has chromatic number $\overline{m}$.

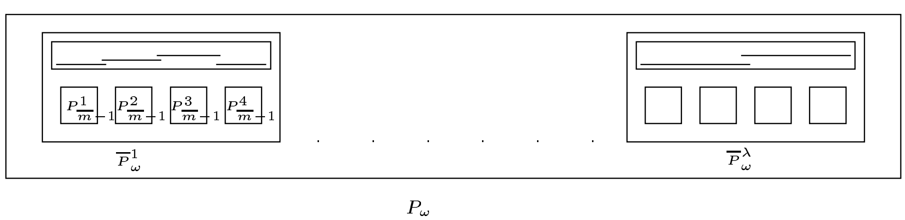


Figure 5.1: The definition of the probability distribution.

5.1.2 The proof of the lower bound

The proof of the lower bound is based on the following lemma:

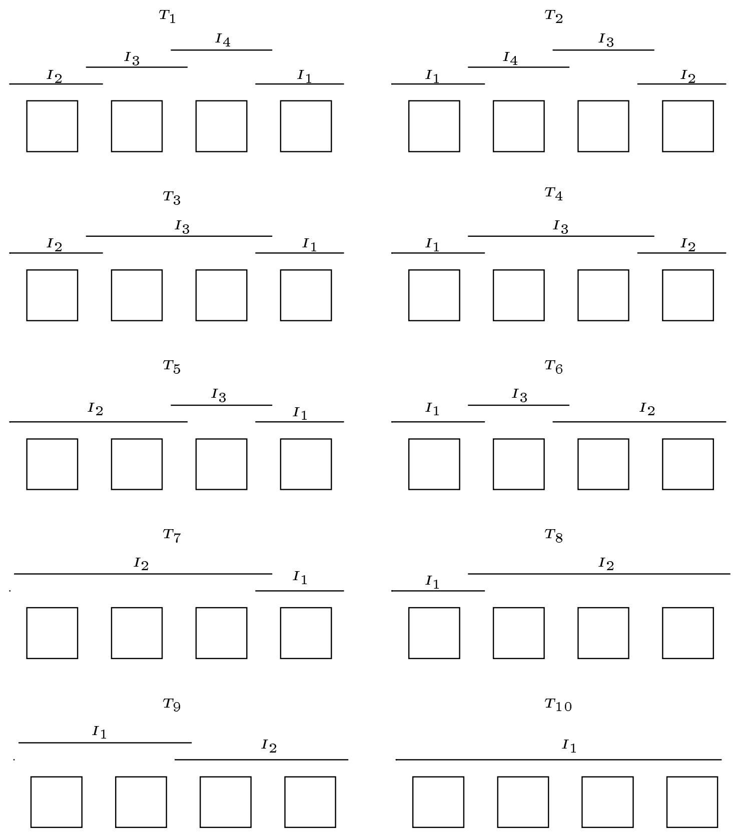


Figure 5.2: The 10 configurations used to form the probability distribution.

Lemma 5.1.1 *Any deterministic online algorithm uses at least $3\omega - 2$ colors with probability at least $1 - e^{-c}$ on an input sequence drawn from probability distribution $P_{\overline{m}}$, if $\lambda \geq \frac{10c}{(1-e^{-c})^4}$.*

From the above lemma, choosing constant c large enough, say $c = \ln 3\overline{m}^3$, we obtain the following Theorem:

Theorem 5.1.2 *For any randomized algorithm for online interval graph coloring, there exists an input sequence of chromatic number ω where the expected number of colors used by the algorithm is at least $3\omega - 2 - o(1/\overline{m})$.*

The remaining part of this section is devoted to the proof of Lemma 5.1.1. The proof is by induction.

The claim of Lemma 5.1.1 holds for $\omega = 1$, since a deterministic algorithm uses one color for the sequence from probability distribution P_1 containing one single interval. Assume the claim holds for a probability distribution $P_{\omega-1}$, i.e. with probability at least $1 - e^{-c}$, the deterministic algorithm uses $3(\omega-1)-2$ colors for an input sequence drawn from a probability distribution $P_{\overline{m}-1}$.

As an intermediate step of the proof we will prove a claim that holds for a probability distribution $\overline{P}_{\omega}^{j}$, $j=1,..,\lambda$. We denote in the following by $\overline{P}_{\omega}$ the generic probability distribution $\overline{P}_{\omega}^{j}$.

Lemma 5.1.3 *Consider a probability distribution $\overline{P}_{\omega}$. Assume the deterministic algorithm uses at least $3(\omega-1)-2$ colors for every input sequence σ^i drawn from probability distribution $P_{\omega-1}^{i}$, $i=1,...,4$. With probability at least $1/10$, the deterministic algorithm uses at least $3\omega-2$ colors for an input sequence drawn from probability distribution $\overline{P}_{\omega}$.*

Proof: Denote by C^i the set of colors used for a generic input sequence σ^i drawn from probability distribution $P_{\overline{m}-1}^{i}$, and let $C_{\overline{m}-1}=\cup_{i=1}^{4}C^i$. Denote by $|C|$ the cardinality of set C. Let $\overline{C}_{\overline{m}}$ be the set of colors used for an input sequence from probability distribution $\overline{P}_{\overline{m}}$. We will prove that $|\overline{C}_{\overline{m}}|\geq 3\omega-2$ with probability at least $1/10$.

We distinguish four cases on the basis of the value of $d=|\cup_{i=1}^{4}C^i|-(3(\omega-1)-2)$, the number of colors exceeding $3(\omega-1)-2$ used by the algorithm for the four sequences σ^i, $i=1,...,4$. We will separately consider the cases of $d=0,1,2$ and $d\geq 3$.

d=0. In this case the deterministic algorithm uses the same set of colors for every sequence σ^i, $i=1,..,4$. The new intervals presented in any of the 10 configurations must be assigned colors not in $C_{\overline{m}-1}$. With probability $2/5$, one of configurations T_1,T_2,T_3 or T_4 is presented. In all these configurations, one among intervals I^1 and I^2 contains all the intervals of σ^4, the other all the intervals of σ^1. We further distinguish two cases: *a.)* Intervals I^1 and I^2 have assigned the same color, say c_1; *b.)* Intervals I^1 and I^2 have assigned different colors, say c_1 and c_2.

a. With probability $1/2$, the sequence is completed by intervals I^3 and I^4 of configuration T_1 or T_2. Interval I^4 must be assigned a color different from c_1, say c_2, since it is overlapping interval I^1. Interval I^3 must be assigned a color different from c_1 and c_2, say c_3, since it is overlapping intervals I^4 and I^2. Then, with probability $1/5$, 3 more colors are used, and the claim is proved.

b. With probability $1/2$, the sequence is completed by interval I_3 of configuration T_3 or T_4. Interval I^3 is assigned a color different from c_1 and c_2, say c_3, since it overlaps both I^1 and I^2. Also in this case, with probability 1/5, 3 more colors are used, thus proving the claim.

d=1. We prove that with probability at least 1/10, 2 new colors are used by the deterministic algorithm. Observe that a sequence σ^i may not use one of the colors of $C_{\overline{m}-1}$ that might be "re-used" for an interval of a configuration T_j overlapping σ^i.

The following simple fact follows since a C^i contains at least $|C_{\overline{m}-1}| - 1$ colors:

Claim 5.1.4 *For any two sequences σ^i, σ^j, $i \neq j$, if $C^i \neq C^j$ then $C^i \cup C^j = C_{\overline{m}-1}$.*

We separately consider 4 different cases distinguished on the maximum cardinality s of a subset $\mathcal{S} \subseteq \{\sigma^1, \sigma^2, \sigma^3, \sigma^4\}$ such that for every $\sigma_i, \sigma_j \in \mathcal{S}$ it holds $C^i = C^j$. We will separately consider the four different cases, $s = 1, 2, 3, 4$.

s=1. In this case, every two sequences have assigned a different set of colors. Then, we have $C^1 \cup C^2 = C^3 \cup C^4 = C_{\overline{m}-1}$. With probability 1/10 configuration T_9 is given. Interval I_1 is assigned a color $c_1 \notin C_{\overline{m}-1}$, since for any color of $C_{\overline{m}-1}$, interval I_1 overlaps an interval assigned that color. For the same reason, a color $c_2 \notin C_{\overline{m}-1}$ is assigned to I_2. Color c_2 must be distinct from c_1 since interval I_2 intersects interval I_1. The claim is then proved.

s=2. The case in which there are at most two sequences assigned with same set of colors is broken in three subcases: A) σ^1 and σ^4 *or* σ^2 and σ^3 *or* σ^1 and σ^3 *or* σ^2 and σ^4 are assigned the same set of colors; B) Both σ^1 and σ^2 *and* σ^3 and σ^4 are the same set of colors; C) σ^1 and σ^2 are assigned different set of colors while σ^3 and σ^4 are assigned the same set of colors. The symmetric of case C. is proved in an analogous way. The 3 cases cover all the possibilities: Case A) excludes cases B) and C); If case A) does not hold then either $C_1 = C_2$ or $C_3 = C_4$ (case C) or both $C_1 = C_2$ and $C_3 = C_4$ (case B).

A Since $s = 2$, we have $C^1 \neq C^2$ and $C^3 \neq C^4$. The same argument of *s=1* applies here to prove the claim.

B) In this case we know that $C^1 = C^2 \neq C^3 = C^4$. With probability 2/5, corresponding to configurations T_1, T_3, T_5 and T_7, interval I_1 includes all the intervals of σ^4. Assume I_1 is assigned a color $c_1 \notin C_{\overline{m}-1}$. With probability 1/4 configuration T_7 is given. For every color of $C^2 \cup C^3 = C_{\overline{m}-1}$, interval I_2 of configuration T_7 overlaps an interval assigned with that color. Since interval I_2 intersects I_1, it is assigned a color $c_2 \notin C_{\overline{m}-1}$ distinct from c_1, then proving the claim.

Otherwise, consider the case in which interval I_1 of configuration T_1, T_3, T_5 or T_7 is assigned a color of $C_{\overline{m}-1}$, say c_1^o. With probability 1/2, corresponding to configurations T_1 and T_3, interval I_2 includes sequence σ^1. Consider the case in which I_2 is assigned a color $c_1 \notin C_{\overline{m}-1}$. With probability 1/2, corresponding to the selection of configuration T_3, interval I_3 overlaps all the intervals of σ^2 and σ^3. For every color of $C^2 \cup C^3 = C_{\overline{m}-1}$, interval I_3 overlaps an interval assigned with that color. I_3 also intersects interval I_2 assigned with a color $c_1 \notin C_{\overline{m}-1}$. Interval I_3 is then assigned a color $c_2 \notin C_{\overline{m}-1}$ thus proving the claim.

We are left to consider the case of I_1 and I_2 assigned with a color of $C_{\overline{m}-1}$, say c_1^o and c_2^o. With probability 1/2, corresponding to configuration T_1, interval I_3 includes sequence σ^2 and interval I_4 includes sequence σ^3. Since $C^1 = C^2$ and $C^3 = C^4$, we have $C^2 \cup c_2^o = C_{\overline{m}-1}$ and $C^3 \cup c_1^o = C_{\overline{m}-1}$. Interval I_3 that overlaps σ^2 and intersects interval I_2, must be assigned a color $c_1 \notin C_{\overline{m}-1}$. Interval I_4, that includes sequence σ^3, and intersects interval I_1 and I_3, must be assigned a color $c_2 \notin C_{\overline{m}-1}$ distinct from c_1. The claim is then proved.

C.) In this case $C^1 \neq C^2$ and $C^3 = C^4$. Thus, from Claim 5.1.4, $C^1 \cup C^2 = C_{\omega-1}$. Since $s = 2$, $C^1 \neq C^3$ and $C^2 \neq C^3$. With probability 2/5, configuration T_1, T_3, T_5 or T_7 is given, and interval I_1 that includes sequence σ^4 is presented. Consider the case in which interval I_1 is assigned a color $c_1 \notin C_{\overline{m}-1}$. With probability 1/4, corresponding to configuration T_7, interval I_2, that includes σ^1, σ^2 and σ^3, is presented. For any color of $C_{\overline{m}-1}$, I_2 includes an interval assigned with that color. I_2 also intersects interval I_1 assigned with color c_1. I_2 is thus assigned a color $c_2 \notin C_{\overline{m}-1}$ different from c_1, thus proving the claim.

We finally consider the case of I_1 assigned with a color $c_1^o \in C_{\overline{m}-1}$. With probability 1/4 configuration T_5 is presented. Interval I_2 overlaps sequences σ^1 and σ^2. Since $C^1 \cup C^2 = C_{\overline{m}-1}$, I_2 is assigned a new color, say c_1. Since $C^3 = C^4$, $C^3 \cup c_1^o = C_{\overline{m}-1}$. Interval I_3 overlaps all the intervals of σ^3, I_1 assigned with color

c_1^0 and I_2 assigned with color c_1. Interval I_3 is then assigned a color $c_2 \notin C_{\overline{m}-1}$, thus proving the claim.

s=3 We have two symmetric cases, either $C_3 = C_4$ or $C_1 = C_2$. If $C^3 = C^4$, we have that either $C^1 = C^3$ or $C^2 = C^3$, but $C^1 \neq C^2$. Under these assumptions, the same argument used in case C.) of *s=2* allows to prove the claim. Case $C_1 = C_2$ is analogous.

s=4 In case $C^1 = C^2 = C^3 = C^4$, all the sequences are assigned the same set of colors. The same analysis of case $d = 0$ allows to prove the claim.

d=2. In this case $|C_{\overline{m}-1}| = 3\overline{m} - 3$. To prove the claim, a new color must be used with probability at least 1/10. With probability 1/10, configuration T_{10} is presented. For any color of $C_{\overline{m}-1}$, interval I_1 overlaps an interval assigned with that color. I_1 is thus assigned a new color $c_1 \notin C_{\overline{m}-1}$.

d≥3. In this case $|C_{\overline{m}-1}| \geq 3\overline{m} - 2$, the claim is then proved.

■

We finally present the proof of Lemma 5.1.1. Let $p(\overline{m})$ be the probability that a deterministic algorithm uses $3\overline{m} - 2$ colors on an input sequence from probability distribution $P_{\overline{m}}$. Consider a probability distribution $\overline{P}^j_{\overline{m}}$ formed by the union of probability distributions $P^i_{\overline{m}-1}$, $i = 1, .., 4$. By induction, we assume that the algorithm uses at least $3(\overline{m} - 1) - 2$ colors with probability $p(\overline{m} - 1) \geq (1 - e^{-c})$ on an input sequence σ^i drawn from probability distribution $P^i_{\overline{m}-1}$.

With probability $p(\overline{m}-1)^4$, a deterministic algorithm uses at least $3(\overline{m}-1)-2$ colors for all the input sequences σ^i drawn from probability distributions $P^i_{\overline{m}-1}$, $i = 1, .., 4$. With probability $p(\overline{m} - 1)^4$ we are then under the assumptions of Lemma 5.1.3. We then obtain from Lemma 5.1.3 that with probability at least $\frac{1}{10}p(\overline{m}-1)^4$, the algorithm uses at least $3\overline{m}-2$ colors for an input sequence from $\overline{P}^j_{\overline{m}}$, $j = 1, .., \lambda$.

Since all the $\overline{P}^j_{\overline{m}}$ that form $P_{\overline{m}}$ are mutually independent, the probability that a deterministic algorithm uses less than $3\overline{m} - 2$ colors on all the input sequences drawn from probability distributions $\overline{P}^j_{\overline{m}}$ is then upper bounded by

$$\left[1 - \frac{1}{10}p(\overline{m} - 1)^4\right]^\lambda \leq \left[1 - \frac{1}{10}(1 - e^{-c})^4\right]^\lambda \leq e^{-\lambda\frac{(1-e^{-c})^4}{10}}.$$

If $\lambda \geq \frac{10c}{(1-e^{-c})^4}$, the given expression has value less than e^{-c}.

We have then proved that with probability at least $(1 - e^{-c})$, a deterministic algorithm uses at least $3\overline{m} - 2$ colors for a sequence from a probability distribution $\overline{P}_{\overline{m}}$, thus implying the claim of Lemma 5.1.1.

5.2 A lower bound for online path coloring on trees

We prove that any randomized algorithm for online path coloring on trees of diameter $\Delta = O(\log n)$ has competitive ratio $\Omega(\log \Delta)$.

We establish the lower bound using Yao's Lemma [94]. We prove a lower bound on the competitive ratio of any deterministic algorithm for a given probability distribution on the input sequences for the problem.

The tree network we use for generating the input sequence is a complete binary tree of $L \geq 4$ levels. The root of the tree is at level 0, the leaves of the tree are at level $L - 1$. The 2^l vertices of level l are denoted by r_j^l, $j = 0, ..., 2^l - 1$. The direct ancestor of vertex u is denoted by $p(u)$. We will indicate by $[a, b]$ the path in the tree from vertex a to vertex b.

The input sequence for the lower bound is generated in $\rho = \Omega(\log L)$ stages. We will prove that at stage $i = 0, .., \rho - 1$, with high probability, the number of colors used by any deterministic algorithm is i. An optimal algorithm is shown to be able to color all the paths of the sequence with only 2 colors, thus proving the lower bound.

At stage i of the input sequence, we concentrate on a specific level $l_i = l_{i-1} - \lceil (3^i \log 16\rho + \log(16(\rho - i) \log n)) \rceil - 1$ of the tree, with $l_0 = L - 1$. It turns out that at least $(16\rho)^{3^i} 16(\rho - i) \log n$ vertices of level l_{i-1} are contained in each subtree rooted at a vertex of level l_i. To simplify notation, the jth vertex of level l_i, $r_j^{l_i}$, is denoted by r_j^i.

We define at stage i a set of pairs $\mathcal{I}_i = \{(u_j^i, v_j^i), j = 0, ..., 2^{l_i} - 1\}$, where u_j^i, v_j^i are two leaves of the subtree rooted at vertex r_j^i of level l_i.

Set $\mathcal{I}_0 = \{(r_j^0, r_j^0), j = 1, ..., 2^L - 1\}$ is composed by one degenerated pair for every leaf of the tree.

Set $\mathcal{I}_i$ is formed by selecting at random for every vertex r_j^i, two distinct pairs of $\mathcal{I}_{i-1}$ in the subtree rooted at r_j^i. The pair associated with vertex r_j^i, in the following often denoted by pair r_j^i, is formed by the two second vertices of the two selected pairs. More formally:

1. For every vertex r^i_j of level l_i, $j = 0, ..., 2^{l_i} - 1$:

   Select uniformly at random two distinct vertices $r^{i-1}_{k^1_j}, r^{i-1}_{k^2_j}$ of level l_{i-1} in the subtree rooted at vertex r^i_j. Let $(u^{i-1}_{k^1_j}, v^{i-1}_{k^1_j}), (u^{i-1}_{k^2_j}, v^{i-1}_{k^2_j}) \in \mathcal{I}_{i-1}$ be the two pairs associated with vertices $r^{i-1}_{k^1_j}$ and $r^{i-1}_{k^2_j}$.

2. $\mathcal{I}_i = \{(v^{i-1}_{k^1_j}, v^{i-1}_{k^2_j}) : j = 0, ..., 2^{l_i} - 1\}$ is the set of pairs at stage i.

The input sequence at stage i is formed for every pair r^i_j of $\mathcal{I}_i$, by a path from the first vertex of the pair to the direct ancestor of vertex r^i_j:

$\mathcal{P}_i = \{[u^i_j, p(r^i_j)] : j = 0, ..., 2^{l_i} - 1\}$.

We prove in the following that any optimal algorithm serves the input sequence with two colors. We first observe:

Lemma 5.2.1 *Every edge of the tree is included in at most two paths of* $\cup_{i\geq 0}\mathcal{P}_i$.

Proof: For every vertex r^i_j, for every stage i, denote by E^i_j the set of edges in the subtree rooted at r^i_j with both endpoints of level between $l_{i-1} - 1$ and l_i, plus the edge $(r^i_j, p(r^i_j))$. (For a leaf vertex r^0_j, E^0_j includes only the edge $(r^0_j, p(r^0_j)))$. We separately prove the claim for every set of edges E^i_j.

Edges of E^i_j are not included in any path $\mathcal{P}_{i'}$, $i' < i$. Every leaf vertex is the endpoint of at most one path in the sequence and one path from $\mathcal{P}_0$. By the construction of the input sequence, vertices u^i_j and v^i_j are the only leaf vertices in the subtree rooted at r^i_j that may be endpoints of paths in a set $\mathcal{I}_{i'}$, $i' \geq i$. The claim is then proved. ■

The following lemma bounds the size of the optimal solution.

Lemma 5.2.2 *The optimal number of colors for any input instance from the probability distribution is 2.*

Proof: We show a two coloring of an input instance with: (i.) Every path directed from a leaf to an ancestor of the leaf; (ii.) Every edge of the tree included in at most two paths. We proceed from the top to the bottom of the tree. Consider an internal vertex v (initially the root), and let v_1 and v_2 be the two children of v. Consider edge (v_1, v). (A similar argument holds for edge (v_2, v)). If no path of the input sequence includes both (v_1, v) and $(v, p(v))$ (assume this is the case if v is the root), edge (v_1, v) is crossed by at most 2 paths, say p_1 and p_2, that end at v. Paths p_1 and p_2 are assigned the two available colors. If only one path, say p_1, includes both (v_1, v) and $(v, p(v))$, there is at most one path, say p_2, including edge (v_1, v) that ends at v. Path p_2 is assigned the color not given to p_1. If there are two paths, p_1 and p_2, including both (v_1, v) and $(v, p(v))$, these have already been colored. The coloring procedure then moves to consider vertex v_1. ∎

In the remainder of the section we show that the expected number of colors used by any deterministic online algorithm is $\Omega(\log L)$, thus implying the lower bound.

The following lemma will be used to prove our result.

Lemma 5.2.3 *For each pair $(u_j^i, v_j^i) \in \mathcal{I}_i$, path $[v_j^i, p(r_j^i)]$ intersects a single path in every set $\mathcal{P}_j$, $j \leq i$.*

Proof: We prove the claim by induction on the number of stages. The claim is true for pairs of $\mathcal{I}_0$. Path $[v_j^i, p(r_j^i)]$ is formed by the union of paths $[v_{k_j^2}^{i-1}, p(r_{k_j^2}^{i-1})]$ and $[p(r_{k_j^2}^{i-1}), p(r_j^i)]$.

If the claim holds at stage $i-1$, path $[v_{k_j^2}^{i-1}, p(r_{k_j^2}^{i-1})]$ intersects one single path for every $\mathcal{P}_j$, $j \leq i-1$.

Path $[p(r_{k_j^2}^{i-1}), p(r_j^i)]$ includes only edges of level lower than $l_{i-1} - 1$. Since no path of a set $\mathcal{P}_j$, $j \leq i-1$, includes edges of level lower than $l_{i-1} - 1$, path $[u_j^i, p(r_j^i)]$ can intersect only path $[u_j^i, p(r_j^i)] \in \mathcal{P}_i$ on edge $(r_j^i, p(r_j^i))$. This is also the single path of $\mathcal{P}_i$ in the subtree rooted at vertex r_j^i, thus showing the claim. ∎

We introduce some more notation. Given a pair $(u_j^i, v_j^i) \in \mathcal{I}_i$, let $C_j^i = \{c_0, ..., c_i\}$ be a set of $i+1$ colors, where c_j is the color assigned to the single path of $\mathcal{P}_j$ intersecting $[v_j^i, p(r_j^i)]$.

Pair (u_j^i, v_j^i) is a *good pair* if C_j^i is formed by $i+1$ distinct colors. We will prove that with high probability, for any stage $i = 0, .., \rho$, there exists at least one good pair of level i. The existence of a good pair of level i gives the evidence that at least $i+1$ colors have been used by the deterministic algorithm, thus proving the claim.

We also denote by ON the number of colors used by the online deterministic algorithm on a specific input sequence drawn from the probability distribution. X_j^i is a boolean variable denoting the event that r_j^i is a good pair.

The following claim gives us a sufficient condition for a pair of level l_i to be a good pair.

Lemma 5.2.4 *Pair* $(u^i_j, v^i_j) \in \mathcal{I}_i$ *of level* i *is a good pair if it is obtained from selecting two good pairs* $(u^{i-1}_{k^1_j}, v^{i-1}_{k^1_j})$, $(u^{i-1}_{k^2_j}, v^{i-1}_{k^2_j})$ *with* $C^{i-1}_{k^1_j} = C^{i-1}_{k^2_j}$.

Proof: For every color $c \in C^{i-1}_{k^1_j} = C^{i-1}_{k^2_j}$, by Lemma 5.2.3, path $[v^{i-1}_{k^1_j}, p(r^{i-1}_{k^1_j})]$ and $[v^{i-1}_{k^2_j}, p(r^{i-1}_{k^2_j})]$ intersect a single path assigned with color c. Path $[u^i_j, p(r^i_j)]$ that contains $[v^{i-1}_{k^1_j}, p(r^{i-1}_{k^1_j})]$ is then assigned with a color $c_i \notin C^{i-1}_{k^1_j}$. Path $[v^i_j, p(r^i_j)]$ intersects path $[u^i_j, p(r^i_j)]$ assigned with color c_i on edge $(r^i_j, p(r^i_j))$. Pair (u^i_j, v^i_j) is then a good pair with set of colors $C^i_j = C^{i-1}_{k^2_j} \cup \{c_i\}$. ■

Let $n_i = (16\rho)^{3^i} 16(\rho - i)\log n$ be the number of vertices of level l_i in a subtree rooted at a vertex of level l_{i+1}. Let $f_i = \sum_j X^i_j$ be the number of good pairs of level l_i in a subtree rooted at a vertex of level l_{i+1}. Let p_{i+1} be a value such that let $\Pr[X^{i+1}_j | ON \le \rho \cap f_i \ge n_i p_i/2] \ge p_{i+1}$, namely a lower bound on the probability of a pair r^{i+1}_j of level l_{i+1} to be a good pair when the algorithm uses at most ρ colors along the sequence and there are at least $n_i p_i/2$ good pairs of level l_i in the subtree rooted at vertex r^{i+1}_j.

The next Lemma assigns a value to p_i.

Lemma 5.2.5 *For a pair* r^{i+1}_j *of level* l_{i+1}, $\Pr[X^{i+1}_j | ON \le \rho \cap f_i \ge n_i p_i/2] \ge p_{i+1} = \left(\frac{1}{16\rho}\right)^{3^{i+1}}$.

Proof: We prove the claim by induction. The claim holds for level l_0 since pair r^0_j is associated with a single color. Assume by induction it holds for level l_i. For the proof of the claim we denote $i_1 = (u^i_{k^1_j}, v^i_{k^1_j})$ and $i_2 = (u^i_{k^2_j}, v^i_{k^2_j})$ the two pairs of $\mathcal{I}_i$ selected at random in the subtree rooted at r^{i+1}_j and by $C^i_{k^1_j}$ and $C^i_{k^2_j}$ the two set of colors associated with the two pairs. In the following we shorten notation by indicating $X_1 = X^i_{i_1}$, $X_2 = X^i_{i_2}$,$C_1 = C^i_{k^1_j}$, $C_2 = C^i_{k^2_j}$.

By Lemma 5.2.4, we can derive the following equality for the probability of a pair of level l_{i+1} to be a good pair:

$$\Pr[X^{i+1}_j | ON \le \rho \cap f_i \ge n_i p_i/2] \quad = \quad \Pr[X_1 \cap X_2 \cap C_1 = C_2 | ON \le \rho \cap f_i \ge n_i p_i/2]$$

Since we are conditioned to those input sequences for which $ON \le \rho$, every pair of level l_i is associated with at most $\frac{\rho!}{(\rho-i)!i!} \le \rho^i$ different set of colors. Let $C \le \rho^i$ be the number of distinct set of colors assigned to the good pairs of level i. To estimate a value for p_{i+1}, we consider the ratio between the cardinality of the set of positive events and the cardinality of the total set of events in the random selection of two pairs at step 1 of the probability distribution. We compute the probability of having a good event conditioned to the existence of at least $n_i p_i/2$ good pairs of level l_i in the subtree rooted at a vertex r_j^{i+1}. The cardinality of the set of positive events is minimized when the good pairs are equally distributed among the C color classes. We therefore obtain:

$$\begin{aligned}
\Pr[X_1 \cap X_2 \cap C_1 = C_2 | ON \leq \rho \cap f_i \geq n_i p_i] &\geq C \frac{\lfloor \frac{mp_i}{2C} \rfloor \cdot \lfloor \frac{mp_i}{2C} - 1 \rfloor}{m(m-1)} \\
&\geq C \frac{(\frac{mp_i}{2C} - 1) \cdot (\frac{mp_i}{2C} - 2)}{m(m-1)} \\
&\geq C \frac{\frac{mp_i}{4C} \frac{mp_i}{4C}}{m^2} \\
&= \frac{1}{16C} p_i^2
\end{aligned}$$

We then apply the inductive hypothesis to obtain

$$\begin{aligned}
\Pr[X_j^{i+1} | ON \leq \rho \cap f_i \geq n_i p_i / 2] &\geq \frac{1}{16\rho^i} \left(\frac{1}{16\rho} \right)^{23^i} \\
&= \left(\frac{1}{16\rho} \right)^{i+23^i} \\
&\geq \left(\frac{1}{16\rho} \right)^{3^{i+1}},
\end{aligned}$$

therefore proving the claim. ■

The claim of the next Lemma is a statement of sharp concentration around the expectation of the number of good pairs at every stage of the probability distribution.

Lemma 5.2.6 $\Pr[f_i \geq n_i p_i / 2 | ON \leq \rho] \geq 1 - \frac{1}{n^{2(\rho - i) - 1}}$.

Proof: We prove the claim by induction. The claim holds for a level l_0 since the single pair of level l_0 is a good pair. Assume Claim 5.2.5 for level l_{i+1} and Claim 5.2.6 for level l_i. We then prove Claim 5.2.6 for level l_{i+1}.

To give a statement of sharp concentration around the expectation of function $f_i = \sum_j X_j^i$ we should consider that random variables X_j's are not necessarily independent since the deterministic algorithm may coordinate the coloring of paths in different subtrees. Therefore, we can use Chernoff-Hoeffding bounds only if we make additional assumptions on the coloring produced by the deterministic algorithm until stage i.

Consider all vertices r_j^{i+1} in a subtree rooted at a vertex of level l_{i+2}. We restrict our attention to those input sequences $\mathcal{I}$ for which the number of good pairs of level l_i in a subtree rooted at every vertex r_j^{i+1} is $x_j \geq n_i p_i/2$, $j = 1, ..., n_{i+1}$. By the induction hypotesis, the total probability of the sequences in $\mathcal{I}$ is at least $p_{\mathcal{I}} \geq \left(1 - \frac{1}{n^{2(\rho-i)-1}}\right)^n \geq \left(1 - \frac{1}{n^{2(\rho-i-1)}}\right)$.

We prove the claim of sharp concentration for every subset of $\mathcal{I}$ that produced the same input sequence up to stage i. Within every such subset, variables X_j^{i+1} are independent, for which we can safely apply Chernoff-Hoeffding bounds to estimate $\Pr[f_{i+1} \geq n_{i+1}p_{i+1}/2 | ON \leq \rho]$. By the theorem of total probability, this implies the claim for the whole set $\mathcal{I}$.

We apply the following version of Chernoff-Hoeffding bounds: Let $f_{i+1} = \sum_j X_j^{i+1}$ and let $\mu = \sum_j \Pr[X_j^{i+1} = 1]$. For every $\delta > 0$,

$$\Pr[f_{i+1} < (1-\delta)\mu] < e^{(-\mu\delta^2/2)}.$$

By the inductive hypothesis given by Lemma 5.2.5, a lower bound over the expected number of good pairs at level l_{i+1} under condition $f_i \geq n_i p_i/2$, is given by $\mu \geq \overline{\mu} = n_{i+1}p_{i+1} = \left(\frac{1}{16\rho}\right)^{3^i} (16\rho)^{3^i} 16(\rho - i - 1)\log n = 16(\rho - i - 1)\log n$. The following expression is easily obtained from the expression of Chernoff's bounds:

$\Pr[f_{i+1} < (1-\delta)\overline{\mu}] \leq \Pr[f_{i+1} < (1-\delta)\mu] < e^{(-\mu\delta^2/2)} \leq e^{(-\overline{\mu}\delta^2/2)}$.

Setting $\delta = 1/2$ we obtain:

$$\Pr[f_{i+1} < n_{i+1}p_{i+1}/2] < exp(-2(\rho - 1 - 1)\log n) = \frac{1}{n^{2(\rho-i-1)}}.$$

We therefore have for the probability of sharp concentration:

$$\begin{aligned} \Pr[f_{i+1} \geq n_{i+1}p_{i+1}/2 | ON \leq \rho] &\geq p_{\mathcal{I}} \Pr[f_{i+1} \geq n_{i+1}p_{i+1}/2] \\ &\geq \left(1 - \frac{1}{n^{2(\rho-i-1)}}\right)^2 \geq 1 - \frac{1}{n^{2(\rho-i-1)-1}}, \end{aligned}$$

thus proving the claim.

■

The construction of the input sequence is repeated until stage $\overline{i} = \rho - 1$ such that $|\mathcal{I}_{\overline{i}}| = 2^{l_{\overline{i}}} \geq (16\rho)^{3^{\overline{i}}} 16(\rho - i)\log n$. Easy computation shows $\rho = \Omega(\log L)$. Lemma 5.2.6 shows that under these assumptions with probability at least $(1 - \frac{1}{n})$, there exist a large number of good pairs of level $l_{\rho-1}$. Then, with probability at least $(1 - \frac{1}{n})$, a set of ρ distinct colors is used by the deterministic algorithm under the condition that $ON \leq \rho$. We therefore obtain:

$$\begin{aligned}
\mathrm{E}[ON] &\geq E[ON|ON > \rho]\Pr[ON > \rho] + E[ON|ON \leq \rho]\Pr[ON \leq \rho] \\
&\geq \rho\Pr[ON > \rho] + \rho(1 - \frac{1}{n})\Pr[ON \leq \rho] \\
&\geq \rho/2,
\end{aligned}$$

where the last inequality follows since $n > 2$. Lemma 5.2.2 states that the optimal solution uses two colors on any of these input sequences. Thus, the lower bound over the competitive ratio of randomized algorithms is given by $\rho/4$. We then conclude with the following theorem:

Theorem 5.2.7 *There exists a $\Omega(\log \Delta)$ lower bound on the competitive ratio of randomized algorithms for online path coloring on a tree of diameter $\Delta = O(\log n)$.*

5.3 Conclusions

In this chapter we have presented the first randomized lower bounds for online interval graph coloring and online path coloring on tree networks. This line of research is aimed to establish if there exists a specific network topology where randomized online algorithms obtain substantially better competitive ratios than deterministic algorithms.

A first open problem is to close the gap between the randomized lower bound for online path coloring on trees and the best deterministic upper bound known for the problem.

The lower bound for path coloring on trees is actually obtained on a 2-colorable graph. This does not preclude the existence of an algorithm that uses $\chi + O(\log \Delta)$ colors for the problem. A second open problem, posed in [19], is to establish a multiplicative lower bound rather than an additive lower bound, i.e. a lower bound on a graph of arbitrary large chromatic number.

Chapter 6

System Architecture for Location Based Services

CDPD [30] has been deployed in North America for providing data services for mobile users. In this chapter, we present a system architecture whose purpose is extracting location information of a CDPD subscriber from the handheld device and making it available to a location server. The location server can then be accessed by an Internet Service Provider (ISP) in order to offer suitable location based services. For example, location information can be exploited by Yahoo yellow pages to offer the closest emergency services to the mobile users. The localization technique we implemented, i.e. the BSIC localization method, allows us to identify the location of a user within the range of a cell. The main idea of this method is that through a suitable protocol, the MSCI protocol, we can obtain from the modem the Base Station Identification Code (BSIC), namely a number that unambiguously identifies the antenna to which the user is currently connected. Since an antenna covers a specific region (i.e a cell), the BSIC also identifies the cell in which the mobile user is currently located. This technique suffers from an inherent lack of *accuracy*, since the dimensions of a cell has a range that goes from some meters to some kilometers. We discuss methods for improving accuracy that are based on analyzing information obtained from multiple cells and using readily available system parameters like received signal strength indicator (RSSI), block error rates (BLER) etc. Furthermore we compare and contrast the accuracy figures of these techniques with those of a global positioning system (GPS) in order to determine the applicability of the different localization techniques. For example, in a metropolitan area the use of a CDPD based localization technique can be sufficient to the

purposes of general directory services. However, if a user wants to be routed to a specific point of interest a GPS is needed since the accuracy derived from a CDPD system is not sufficient.
We stress that we investigate a *handset assisted solution* to localize a subscriber. This approach, in contrast to network oriented localization solutions, has a minimal impact on the telecommunication network and protects privacy (the user has to explicitly disclose his localization information).
The rest of the chapter is organized as follows. In section 6.1 we describe the current localization techniques and their main features. In section 6.2 we introduce the CDPD architecture, namely the wireless environment in which we developed our localization method. Section 6.3 describes the Modem Status Configuration Interface (MSCI) protocol and the BSIC localization method, while section 6.4 presents a system architecture that allows to map the BSIC into geographic coordinates, namely latitude and longitude. Finally we show how these coordinates can be exploited by Service Providers to offer location based services. In section 6.5 we evaluate the quality of the accuracy of the BSIC localization method, and we compare it to the the GPS accuracy. Finally section 6.6.2 presents two methods, the multiple cells method and the RSSI measurement method, in order to improve the accuracy of the BSIC localization method.

6.1 Localization Technology

The ability to locate the position of a mobile device is crucial for providing geographically specific value-added information, and it has been indicated in a recent study [56] as a key factor for the development of new wireless application.

Applications using mobile location service technologies include fleet management, vehicle tracking for security, tracking for recovery in event of theft, telemetry, emergency services, location identification, navigation, location based information services and location based advertising. The largest push for localization technology is coming from the US. There, mobile telephone operators have been forced by the FCC to provide emergency 911 services by October 2001 in such a way that the location of the caller could be determined within a radius of 125 meters in 67% of all cases. There are three major localization techniques [83]:

- *Triangulation* can be done via *lateration*, which uses multiple distance measurements between known points, or via *angulation*, which measures angle or bearing relative to points with known separation;

- *Proximity* measures nearness to a known set of points;
- *Scene analysis* examines a view from a particular vantage point, such as antennae or mobile terminals.

Implementation of location systems generally uses one or more of these techniques to locate objects and/or people. Furthermore location systems can be either *terminal based* or *network based*. In terminal based systems, it is the mobile device itself that determines the location. Normally these systems are less accurate but they do not require significant network upgrade. Network based systems provide more accurate localization information, but require a significant network upgrade. In the following we summarize the main features of the current localization techniques, see [43, 56] for further details.

- *GPS* GPS (Global Positioning System) is a system that consists of 24 satellites circling the earth in a particular constellation to each other so that several satellites fall within line of sight for any GPS receiver on Earth. Because the satellites are continuously broadcasting their own position and direction, the GPS receiver can calculate its position with an excellent approximation. Anybody can use the GPS system for free with an appropriate receiver. GPS has been developed in the US for military use, but from the beginning of the decade it has been usable (with lower resolution) for civilian purposes. GPS technology for mobile phones is being currently developed for example by SnapTrack and SiRF and it is already used in Benefon dual mode GSM/GPS handsets.
- *TOA* This method uses the Time Of Arrival (TOA) of signals between the mobile phone and the cellular antenna. TOA is used to capture time difference of arrival information to make calculations to determine an estimate of the mobile terminal position. This technology requires large network modifications and is therefore not cost-effective. Rolling out TOA for an entire network is estimated to cost as much as 10 times the price of an E-OTD system.
- *E-OTD* The E-OTD (Enhanced Observed Time Difference) system works by using the existing GSM infrastructure to determine the mobile phones location. When a user calls selected service providers, E-OTD simultaneously sends data indicating the phones position. It works by comparing the relative times of arrival, at the handset and at a nearby fixed receiver, of signals transmitted by the underlying mobile network base stations. The

E-OTD system overlays the existing mobile network. Suppliers for E-OTD solutions include CPS, Ericsson and BT Cellnet.

- *COO* COO (Cell Of Origin) can be used as a location fixing scheme for existing customers of network operators, but it is not as exact as the other methods. It requires no modification to the mobile terminal, but normally the network operator has to do some significant upgrade work. In urban areas COO might be sufficient to determine location fairly accurately, because the cell size is very small. In rural areas, where the cell radius is larger, it might not be enough accurate. The solution we present in this work is based on this approach, but as we will see, it does not require any significant network upgrade.

6.2 CDPD Architecture

The CDPD architecture consists of two essential entities shown in Figure 6.1, the mobile end station (MS) and the network. A user accesses CDPD network using a Mobile end System (MS) that establishes a connection to a Mobile Data Base Station (MDBS) on a specific radio channel. The MDBS is responsible for providing the data link for the set of radio channels serving a cell, i.e. the geographic area covered by a MDBS. A MDBS is controlled by a single Mobile Data Intermediate System (MDIS). MDISs are networked with other MDISs and fixed networks through routing and relay elements known as Intermediate Systems (IS) that are unaware of mobility.
The CDPD forum has specified standards for implementing a Location Service [30]. However, all proposals are network oriented and heavily impact on the network traffic because of location requests. In the following we present our solution, the *BSIC localization method*. We exploit the ability of a MS to get location information from the modem, without requiring any significant network upgrade.

When a CDPD user is connected to a MDBS, the modem is continuously updated with the information on which cell tower the MS is currently connected to. In particular each cell, is identified by a unique number, the Base Station Identification Code (BSIC). The MS can get information on the current BSIC from the modem through the Modem Status Configuration Interface (MSCI) protocol (see section 6.3). Finally, the BSIC can be mapped into geographical coordinates as appropriate for the provision of location-based services.

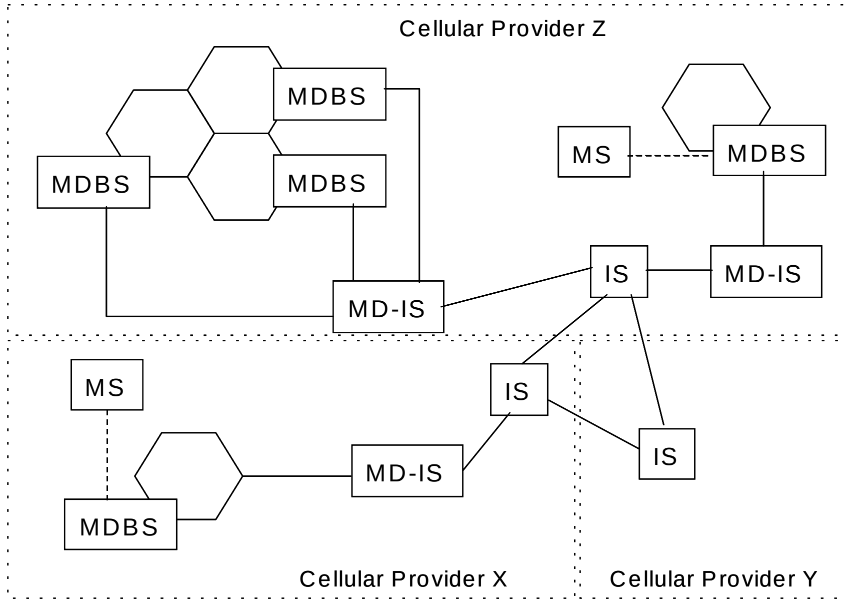


Figure 6.1: CDPD Architecture

For the purposes of a demonstrator, a Compaq handheld Pocket PC (HP3600) was used equipped with a CDPD Novatel Wireless modem MSCI enabled [91], and AT&T Wireless was used as the service provider. Note that for all practical purposes, any modem can be interrogated via the standard serial interface using AT commands. This may however require additional device drivers to be written to enable simultaneous status request and maintain a TCP connection. The MSCI protocol alleviates this requirement.

6.3 The MSCI Protocol

MSCI (Modem Status Configuration Interface) is a protocol which allows the modem parameters to be configured and monitored using UDP packets over a data connection. The MSCI Communication model is based on the client-server paradigm (see Figure 6.2). The client sends a datagram to the server, specifying the function the MSCI Server has to perform and possibly some arguments, and waits for the answer (see Figure 6.3). The MSCI Server is implemented in the Novatel wireless modem and it is identified by the private non-routable IP address 10.0.0.1 on port 4950(see [91] for further details). Therefore the MSCI Server runs on the same handset in which the Novatel wireless modem is installed. Three different kinds of functions are supported by the MSCI protocol:

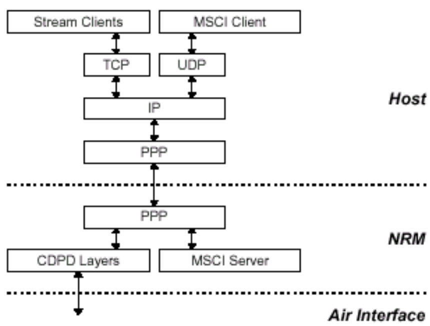


Figure 6.2: The MSCI Communication Model.

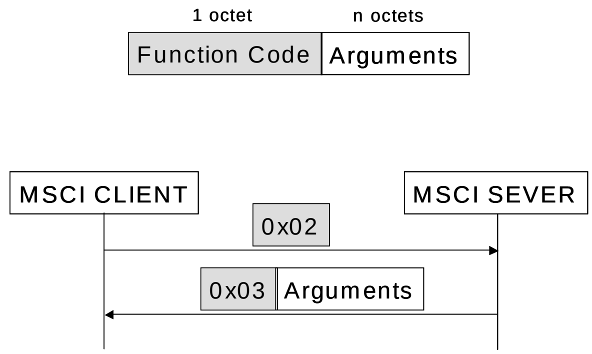


Figure 6.3: Client/Server Messages.

- *Status information:* it allows to gather information on the status of the modem. For example we can obtain the signal strength indicator (RSSI), the BSIC or the block error rate (BLER);

- *Configuration information:* it allows to gather information on the modem configuration. For example we can verify if the modem is configured to detect some specific events;

- *Commands:* it allows to send command to the modem. For example we can properly set the modem registers.

As we have seen in section 6.2 the BSIC unambiguously identifies a cell, hence knowing the current BSIC, namely the BSIC of the base station to which the user is currently connected, we can estimate with some approximation (see Table 6.2) the location of a user. The current BSIC can be gathered invoking the Status Request Function (Function Code 0x02). This function is used by the MSCI client to request the modem status information to the MSCI Server; observe that no arguments are required. The MSCI Server responds with a Status Response (Function Code 0x03). In table 6.1 we show the arguments of the Status Response function. We just consider those fields that we will use to implement our localization methods. The *CDPD Status* field indicates if the channel is acquired and/or the link is stablished and/or the CDPD device is registered. *Current Channel* indicates the channel that is currently used for the communication between the wireless devices and the antenna. *Current RSSI* indicates the current signal strength and finally *Current Cell Site ID* is the BSIC.

Field	Octets	Description
CDPD Status	1	Mask containing 3 bits 0x01: TRUE if channel acquired 0x02: TRUE if link established 0x04: TRUE if registered
Link Status	1	Contains same information as bit 0x02 in the CDPD Status field
Last Registration Error	1	(Block ID 0x40, Parm ID 0x11)
RRM State	1	(Block ID 0x40, Parm ID 0x18)
Current Area Color Code	1	(Block ID 0x40, Parm ID 0x16)
Current Channel	2 MSB first	(Similar Block ID 0x40, Parm ID 0x14)
Current RSSI	1	(Block ID 0x40, Parm ID 0x12)
Current BLER	1	(Block ID 0x40, Parm ID 0x13)
Reserved	1	
Current Power Level	1	(Block ID 0x40, Parm ID 0x19)
Current Service ID	3*2 MSB first	(Similar Block ID 0x40, ParmID 0x17) SPI: 0-65535 WASI: 0-65535 SPNI: 0-65535
Current Cell Site ID (BSIC)	2 MSB first	(Similar Block ID 0x40, Parm ID 0x15)
Current Power Product	1	(Block ID 0x40, Parm ID 0x1a)

Table 6.1: Arguments of the Status Response function

6.4 The GEOPROXY Architecture

The BSIC can be effectively used to estimate the approximate location of a MS. The ability to localize a user just depends on a successful connection to a cell tower.

It remains to explain how this information can be exploited by a service provider. We need a further component that is capable of communicating to a service provider the approximate location in a suitable format, possibly latitude and longitude (BSIC could be meaningless). This is the main goal of the Geoproxy architecture. The overall architecture consists of 5 main components:

- Browser: the user accesses the localization service by a Browser; it may be Netscape or Internet Explorer.

- GEOPROXY: A Web server that maps BSIC into geographical coordinates, namely latitude and longitude. The mapping function could have been implemented inside the MS, without requiring an HTTP connection. Nevertheless we implemented the HTTP based solution for two main reasons: 1) The BSIC label could change over time, 2) Such a solution allows us to send back a customized HTML page to the user (see step 6 in figure 6.4).

- LOCALPROXY: is a local daemon, running on the mobile user device, that acts as an MSCI Client. The Localproxy gets the current BSIC from the modem and it is also responsible for communicating the BSIC to the GEOPROXY. When a user wants to access a localization service, he has to connect by the browser to the Localproxy (just by typing the URL *http://localhost:4444)*.
- The MSCI Server. Recall that the MSCI Server runs on the same MS in which is installed the Novatel wireless modem.
- Content/Service Provider: it exploits the user geographical coordinates to provide location based services.

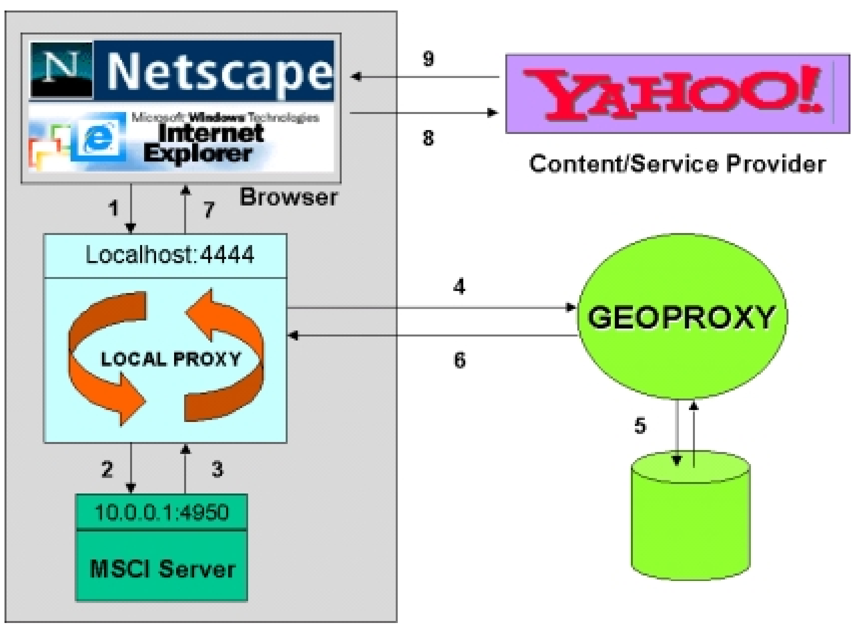


Figure 6.4: The Architecture

In what follows and in Figure 6.4 we outline a description of the procedures to fulfill a new request for a location based service:

1. A user wants to access a location based service. He connects by browser to the LOCAL-PROXY at the URL *http://localhost:4444*;

2. LOCALPROXY acts as a MSCI Client. It sends a Status Request to the MSCI Server;

3. The MSCI Server answers by the Status Response function. LOCAL PROXY gets the BSIC from the arguments of the Status Response;

4. LOCALPROXY sends the BSIC to the GEOPROXY using the HTTP GET (e.g. *http://<GEOPROXY>?BSIC=<BSIC>*);

5. GEOPROXY maps the<BSIC> into geographical coordinates <geo coo>, namely latitude and longitude of the base station;

6. GEOPROXY sends back to the LOCALPROXY the Localized Service Page, an HTML page with links to the Content/Service Provider; each link provides the <geo coo> to a Content/Service Provider through an HTTP GET;

7. LOCALPROXY redirects the incoming HTML page to the Browser which displays the Localization Service Page to the user;

8. The user clicks on a link and connects to a Content/Service Provider;

9. The Content/Service Provider processes the <geo coo> and provides the required service.

6.5 Accuracy

BSIC localization method can be used to provide the approximate location of a user; the main advantages of this technique is that it is easy to implement, it does not require significant network upgrade and it is relatively fast in detecting the BSIC and subsequently perform a translation of the BSIC into geographic coordinates. The main disadvantage, is the lack of accuracy; the dimension of a cell vary from some meters to some kilometers (see Table 6.2).

In some cases and for some applications that accuracy could be considered insufficient. For this reason, in the next sections we try to better evaluate the accuracy of the BSIC localization method, and we compare it with the GPS, at present the best location technology. Finally, we experiment two methods to improve the BSIC localization method.

Cell type	Antenna location	Cell Dimension (km)
Large macrocell	Above rooftop level	3-30
Small macrocell	Above rooftop level	1-3
Microcell	Below or about rooftop level	0.1-1
Picocell	Below rooftop level	0.01-1
Nanocell	Below rooftop level	0.01-0.001

Table 6.2: Different Cell Types

6.5.1 GPS measurement

Using the GPS it is possible to accurately measure the position of a MS. Although the GPS suffers from some inaccuracy, (accuracy mainly depends on the number of satellites visible in a particular location) it is normally limited within 10 meters; hence, it is much more accurate than localization techniques based on cell site boundaries. For this reason, in what follows we refer to the GPS position being the actual location of the mobile user.

To compare the performances of the localization technique based on the BSIC with respect to the GPS, we modified the LOCALPROXY to record every 2 seconds all modem information, namely CDPD status, Received Signal Strength Indication (RSSI), Block Error Rate (BLER), power level and BSIC. Furthermore, for each record in the log, we append the GPS information , namely location coordinates, accuracy (estimated horizontal/vertical position error in meters), velocity and a time stamp. It may happen that some records in the log file miss the modem information and/or the GPS information. This is mainly due to insufficient channel availability for CDPD connections or to shadow effects in the presence of obstacles (buildings or trees) preventing a GPS connection. In the following we denote by *Total Samples* the overall number of records in the log file, while we denote by *Useful Samples* the number of records in the log file for which we have both GPS and modem information. Finally, we denote by *NO GPS* the number of records in the log file for which we do not have GPS information or it is useless and we denote by *NO modem* the number of records in the log file for which we do not have modem information or it is useless (see Figure 6.7). The analysis of the log files allows us to compare the BSIC method with the GPS. In Figure 6.5 we give a pictorial representation of some data in the log files obtained while driving about New York. Each point in the map represents a location obtained through the GPS. Different colors of the points correspond to different BSICs.

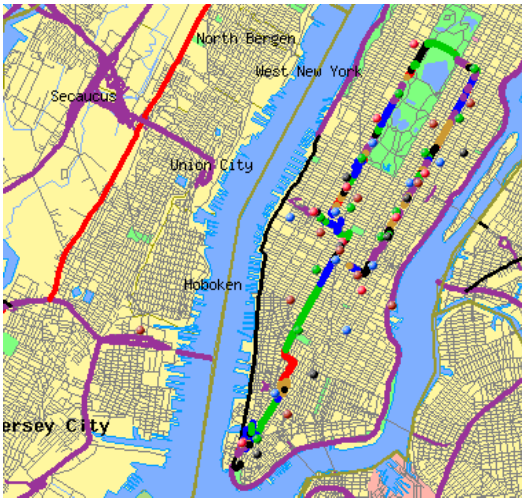


Figure 6.5: A NY BSCI/GPS map

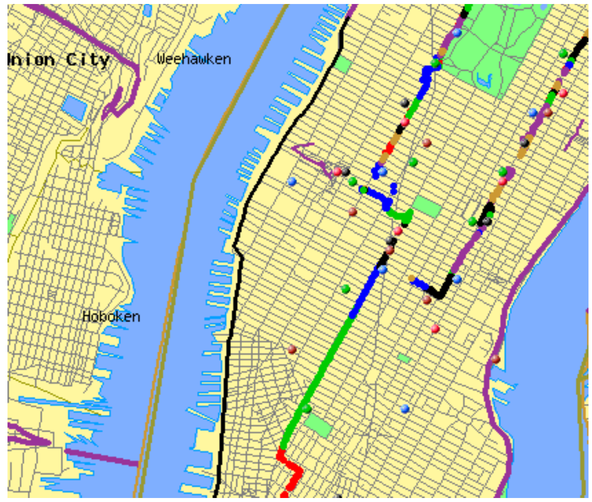


Figure 6.6: A zoom of the NY map

The isolated points in the map, namely the points that are not part of the path, represent the location of the MDBS. Zooming the map (see Figure 6.6) one can see that sometimes the path is not continuous, this is due to lack of GPS connection.

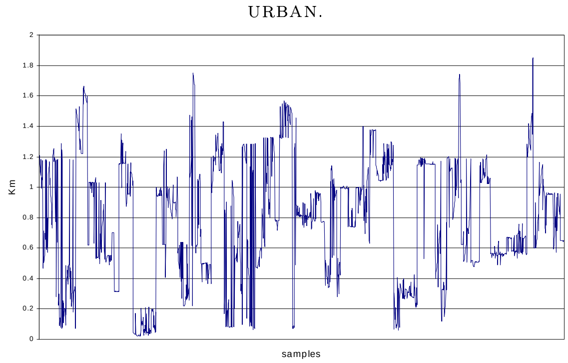
6.5.2 Average distance

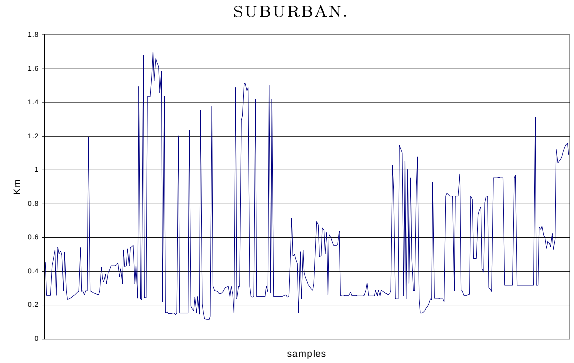
A first measure to estimate the quality of the BSIC localization method is the computation of the average distance of the position of a MS with respect to the correspondent MDBS.

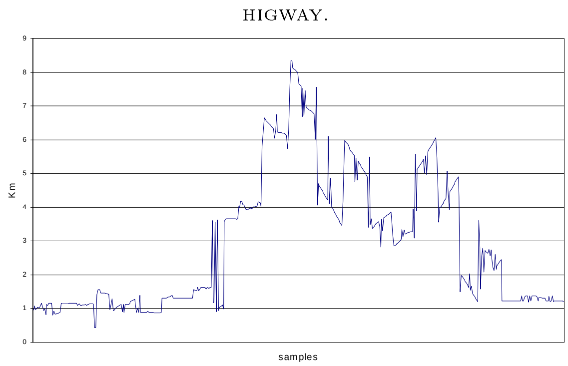
Namely if the log file contains N Useful Samples, and we denote by $d(MDBS^{(i)}, GPS^{(i)})$ the distance between the MDBS (identified by the BSIC in the log file) and the MS (the exact location of the MS is obtained through the GPS) of the ith sample, the average distance is defined as $\Delta d = \frac{1}{N}\sum_{i=1}^{N} d(MDBS^{(i)}, GPS^{(i)})$.

As we have seen in the previous sections, the BSIC localization method suffers from an inherent inaccuracy. We can identify the region in which a MS is located, but not its actual position. Hence, using this localization method, we implicitly assume the likely location of a MS to be the location of the MDBS . Thus a small Δd, with a relatively small variance, gives the evidence that the MDBS location can well approximate the MS actual location in most of the cases. However, a large average distance and/or a large variance can lead to non significant figures.

We ran the experiment in three different contexts, an urban context (NY metropolitan area), a sub-urban context (Madison, NJ-Morris Town, NJ) and finally on a highway context (driving







	URBAN	SUBURBAN	HIGWAY
Total Samples	2237	665	1188
Useful Samples	1840	378	520
Δd	0.79Km	0.49Km	2.91Km
Stand. Deviation	0.39Km	0.38Km	2.00Km
Min.	0.02Km	0.11Km	0.95Km
Max.	1.85Km	1.70Km	8.35Km
No GPS	136	0	68
No modem	261	287	600

Figure 6.7: Average distance experiments.

from NY to Madison, NJ).

The results of the experiments are shown in figure 6.7 . In the table *Min* is the minimum distance of a MS to the corresponding MDBS, namely $min(d(MDBS^{(i)}, GPS^{(i)})), \forall i$, *Max* is the maximum distance of a MS to the corresponding MDBS, namely $max(d(MDBS^{(i)}, GPS^{(i)})), \forall i$.

In the urban area most of the time we have both CDPD and GPS data available (Useful Samples are about 82% of the Total Samples). The average distance is about 800 meters. The standard deviation is about 50% of the average distance and about 21% of the maximum distance. In the suburban area, the average distance (about 500 meters) is even smaller than in NY samples, but the number of useful samples decreases significantly (about 56% of the Total Samples). Furthermore the number of useless CDPD samples is a big fraction of the total. The standard deviation is about 77% of the average distance (the value are more spread with respect to the urban case) and about 22% of the maximum distance. Finally in the highway, the average distance is very high, about 3 kilometers. This can be explained with the presence of big areas without obstacles in the middle between the MT and the MDBS; furthermore higway cell size is bigger than urban cell size. The useful samples are about 44% of Total Samples and the standard deviation is about 69% of the average distance and about 24% of the maximum distance. Finally observe that the coverage of the CDPD network is better in urban area (only about 12% of the total samples have useless modem information, while in suburban and higway areas is about 50%), while the GPS coverage is better in suburban and higway areas (this is mainly because of the presence of high buildings in the urban area).

6.5.3 Service Accuracy

For specific contexts, even a poor quality in accuracy of location method can be sufficiently reliable to offer suitable services to the users. An average distance of 2 Kms can be considered inappropriate in an urban context, but it may be sufficient in a highway context. For this reason we introduce the notion of service accuracy. Suppose to know both the MS's actual position, denoted by $\overline{GPS}$, and the BSIC of the cell tower to which the MS is connected, denoted by $\overline{BSIC}$. Denote by $S_{\overline{GPS},d}$, the set of providers of service S within d Km to $\overline{GPS}$ and by $S_{\overline{BSIC},d}$, the set of providers of service S within d Km to $\overline{BSIC}$. The service accuracy A^* is a function of the service S, the distance d, $\overline{BSIC}$ and $\overline{GPS}$, defined as follows:

$$A^*(S, d, \overline{BSIC}, \overline{GPS}) = |\frac{S_{\overline{BSIC},d} \cap S_{\overline{GPS},d}}{S_{\overline{GPS},d}}|$$

If we call "good service" the services that are close to user actual location, namely $S_{\overline{GPS},d}$, this function counts the fraction of "good services" that we can locate using the $\overline{BSIC}$ as an approximate location of the user.

An alternative measure of accuracy can be the fraction of "approximate services", namely the fraction of services in $S_{\overline{BSIC},d}$ that are also good services. Thus we introduce the following alternative definition of accuracy. The service accuracy A^{**} is a function of the service S, the distance d, $\overline{BSIC}$ and $\overline{GPS}$, defined as follows:

$$A^{**}(S, d, \overline{BSIC}, \overline{GPS}) = |\frac{S_{\overline{BSIC},d} \cap S_{\overline{GPS},d}}{S_{\overline{BSIC},d}}|$$

Both measures provide acceptable and complementary estimations of service accuracy and we refer to both in this work. If a user wants to know which are the closest cafeterias with respect to his location, a service accuracy close to one means that there are no significant differences between the BSIC localization method and the GPS. Nevertheless, if the user wants to know how to get to those cafeterias, the GPS may become essential. Moreover, a good "cafeterias" accuracy does not necessarily imply a good accuracy for other services. In fact, service accuracy

Business Name	Address	City	Phone
Starbucks Coffee Co	684 Avenue Of The Americas	New York, NY	(212) 691-1948
Starbucks Coffee Co	124 8th Ave	New York, NY	(212) 462-2020
Starbucks Coffee Co	93 Greenwich Ave	New York, NY	(212) 462-4697
Starbucks Coffee Co	10 Union Sq E	New York, NY	(212) 982-8426
Starbucks Coffee Co	304 Park Ave S	New York, NY	(212) 475-9025
Starbucks Coffee Co	378 Avenue Of The Americas	New York, NY	(212) 477-2690
Starbucks Coffee Co	261 5th Ave	New York, NY	(212) 779-1557
Starbucks Coffee Co	13 Astor Pl	New York, NY	(212) 982-3563
Starbucks Coffee Co	370 7th Ave Fl 1	New York, NY	(212) 967-8463
Starbucks Coffee Co	424 Park Ave S	New York, NY	(212) 725-0637
Starbucks Coffee Co	51 Astor Pl	New York, NY	(212) 677-6447
Starbucks Coffee Co	334 5th Ave	New York, NY	(212) 279-2960
Starbucks Coffee Co	450 7th Ave Fl 1	New York, NY	(212) 279-1122
Starbucks Coffee Co	395 3rd Ave	New York, NY	(212) 686-2483
Starbucks Coffee Co	141 2nd Ave	New York, NY	(212) 780-0024

Figure 6.8: Yahoo yellow pages.

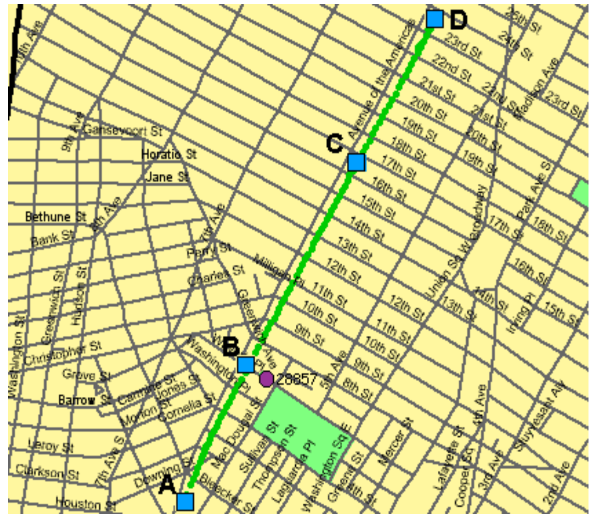


Figure 6.9: Service Accuracy.

mainly depends on the difference between the distribution of specific services around the MDBS and around the actual MS location.

In the following we present some experiments on service accuracy. In Figure 6.9 we have collected the samples driving on Avenue of the Americas (NY) approximately between Bleecker St. and 24 St., while the CDPD modem was continuously connected to the cell tower located at latitude 40.733 and longitude -73.999 (BSIC 28857). The average distance of these samples is about $0.63Km$, sufficiently close to the average distance found in the urban area experiment, see Figure 6.7, to consider this set of samples representative of the urban context. We limit our attention to the samples A,B,C,D. Points A and D are located at the end point of the path, far away from the cell tower, thus they possibly represent a bound to the service accuracy. Point B is the closest point with respect to the cell tower. Point C has a distance (0.77Km) very close to the average urban distance (0.79Km), hence it can be considered as "the average sample" (see Figure 6.7 and Table 6.3).

In order to determine the set of providers $S_{x,d}$ that allow us to calculate the service accuracy, we connect to Yahoo yellow pages as follows:

http://yp.yahoo.com/py/ypResults.py?stx=starbucks&city=New+York&state=NY&
country=us&slt=40.733&sln=-73.999,

where *stx* is the service provider (Starbucks), *slt* is x's latitude and *sln* is x's longitude. The result are the Starbucks cafeterias close to location x (see Figure 6.8).

	latitude	longitude	Distance to the Cell Tower (Km)
A	40.729359	-74.002235	0.49
B	40.733372	-73.999657	0.07
C	40.739378	-73.995312	0.77
D	40.743443	-73.992378	1.29

Table 6.3: Distances.

S=Starbucks $\overline{BSIC}$=28857				
	d			
$\overline{GPS}$	4.82 Km	3.22 Km	1.6 Km[a]	0.8 Km
A	80%	80%	84.6%	62.5%
B	100%	100%	100%	100%
C	80%	90%	92.3%	50%
D	65%	65%	53.8%	12.5%

Figure 6.10: Starbucks Accuracy (A^*)

[a]1 mile

S=Starbucks $\overline{BSIC}$=28857				
	d			
$\overline{GPS}$	4.82 Km	3.22 Km	1.6 Km[a]	0.8 Km
A	80%	80%	91.6%	83.3%
B	100%	100%	100%	100%
C	80%	90%	60%	57.1%
D	65%	65%	35%	11.1%

Figure 6.11: Starbucks Accuracy (A^{**})

[a]1 mile

In Figures 6.10 and 6.11 we show the result of the service accuracy calculation for different values of the distance d.

First observe that the figures of both A^* and A^{**} are very close. Service accuracy is greater than 50% independently from the distance d, except for the farthest point D. In particular point C, that represents the average sample, has a service accuracy greater than 50% with $d = 0.8$ Km. We can conclude that in most cases, half of the Starbucks cafeterias that are considered close using the MS actual position (GPS location) can be considered close also using the BSIC localization method. Observe that point B, being so close to the Cell Tower, has a service accuracy of 100% all the times. Moreover, in almost all cases service accuracy increases with the distance d, but this is not always true. For example if we consider point C (see Figure 6.10), service accuracy is 90% if d is 3.22 Km and decrease to 80% when d is 4.82 Km. This can be explained with the distribution of the Starbucks cafeteria around the MDBS and around point C that strongly influences the accuracy values.
In Figures 6.12 and 6.13 we evaluate the service accuracy of Barnes & Noble bookshops.

S=Barnes & Noble $\overline{BSIC}$=28857				
	d			
$\overline{GPS}$	4.82 Km	3.22 Km	1.6 Km[a]	0.8 Km
A	92%	80%	100%	25%
B	100%	100%	100%	100%
C	92%	100%	100%	75%
D	92%	100%	100%	75%

Figure 6.12: Barnes & Noble Accuracy (A^*)

[a]1 mile

S=Barnes & Noble $\overline{BSIC}$=28857				
	d			
$\overline{GPS}$	4.82 Km	3.22 Km	1.6 Km[a]	0.8 Km
A	100%	100%	83.3%	100%
B	100%	100%	100%	100%
C	100%	90.9%	83.3%	75%
D	92.3%	90.9%	83.3%	75%

Figure 6.13: Barnes & Noble Accuracy (A^{**})

[a]1 mile

Also in this case the figures of both A^* and A^{**} are very close and service accuracy is considerably high in each point also for small values of d. The only exception is the small value of service accuracy of point A when d is 0.8 Km (see Figure 6.12) . This can be explained with a relatively high concentration of Barnes & Noble bookshops north of the cell tower.
We did not extend the evaluation of service accuracy in suburban area, or in the highway, since in these contexts the diffusion of specific service providers like Starbucks or Barnes & Noble is considerably smaller. This means that a query like "which are the closest Barnes & Noble bookshops" may probably have as an answer that the closest bookshop is within ten kilometers. With such big distances there are no significant differences between BSIC location method and GPS, unless the question is "How to get to the closest Barnes & Noble bookshop".

6.6 Techniques to improve BSIC method accuracy

6.6.1 Accuracy improvement using multiple cells

The BSIC method allow us to state that the MS is somewhere inside a specific region covered by a cell, however it is not even possible to determine if the user is located north or south of the MDBS. We simplify and consider the region covered by a cell to be a circle having the center in the MDBS and a fixed radius (from some meters - nanocells, to some kilometers - large macrocell, see Table 6.2). By using multiple cells one can greatly enhance the accuracy of localization. The main idea is as follows: if a MS can connect to multiple cells, this means that the user is in a region covered by all these cells (assuming overlapping cell boundaries). In what follows we discuss in more detail the algorithm used for the *multiple cells localization method*.

The MSCI protocol allows us to specify the *Channel List* (CL), the list of channels to be used for a CDPD connection. If the modem is configured to work in the Use Channel List mode (see MSCI protocol Block ID 0x00 Parm ID 0x05) the connection is established only to those channels specified in the CL. Each MDBS has a set of channels that can be used to establish a CDPD connection. The CL allows us to explore this set of channels identifying all the MDBSs around a MS. The algorithm, is shown in Figure 6.14

1. Set Set use CL mode
2. Set CL = all channels
3. While CL is not empty
4. Start connection
5. The modem connects to MDBS identified by BSIC using channel C
6. Delete C from CL
7. End While

Figure 6.14: The Algorithm

Since each MDBS M uses only a specific set of channels S, when at step 6 we delete all the elements of S from CL, we avoid the reconnection to M. Hence the next possible connection at step 4 will be to another MDBS or possibly to nobody else. Running this algorithm, and possibly exploring the whole CL, we can identify all the MDBSs "visible" to a MS.

Figure 6.15: A multiple cells experiment

InTable 6.4 we show the results of an experiment on multiple cells localization method. The

Round	BSIC	MDBS Location	Channel
1	33487	C	674
2	33489	C	677
3	33489	C	667
4	6044	Unknown Location	724
5	6044	Unknown Location	725
6	31406	A	685
7	31406	A	681

Table 6.4: Running the algorithm

MS is located in B (see Figure 6.15), and runs the algorithm. During the first three connections the MS connects to a MDBS located in C (NOTE: BSICs 33487, 33488 , 33489 correspond to the same location). At this moment we can only assert that the MS is somewhere in the C region. The fifth and sixth connections are to a cell tower of which we do not know the location. Finally the last two connections are to the BSIC 31406 corresponding to location A. Since the MS was able to connect to both the cell towers A and C, we can conclude that it is somewhere in the region covered by both of them.

Although this method can be considered as an improvement of the BSIC method, it has several disadvantages. The algorithm is extremely slow. Performances can be improved by knowing the association between channels and cells in advance, in this way step 6 of the algorithm could delete, in just one step, all channels associated to a particular MDBS, avoiding the reconnection to the same cell more than once. Furthermore if the algorithm connects to just one cell, this does not necessarily mean that this is the only cell visible by the user. In fact it may happen that the connection is not allowed due to traffic congestion. Finally, any insertion/deletion from the CL requires a shutdown and restart of the modem, and consequently any data flow would be interrupted. Anyway we think that most of these problems can be addressed by simple improvements of the MSCI protocol.

6.6.2 Accuracy improvement using RSSI

RSSI measurements can be exploited to improve accuracy of location [90]. Several methods have been studied to estimate the distance through RSSI [52], but all of them suffer from inaccuracy

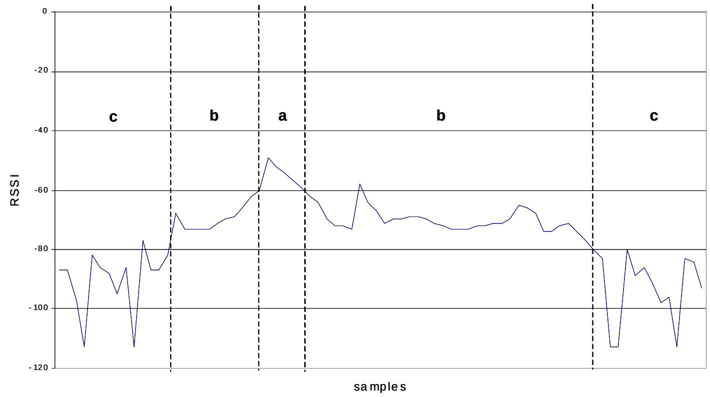


Figure 6.16: RSSI measurement driving on route A

due to the differences in terrain attenuation. The main problem is that the same value of RSSI can correspond to dramatically different distances due to the presence of obstacles between the MS and the MDBS. Jean-Marc Latapy in [50] proposes a new model to determine the distance, based on the construction of the RSSI interval map. This map identifies the regions around a MDBS with values of the RSSI in the same interval. In other words as an isobaric map shows same pressure layers, and can be used to identify areas with specific pressure levels, an RSSI interval map shows layers with the same RSSI. If we know the value R of the current RSSI, and the RSSI interval map around the MDBS, then we can identify the interval in the map containing R, and thus we can better estimate the region in which the MS is located.

The experiments in Figures 6.16 and 6.17 show how to build a simple RSSI map of the region shown in Figure 6.18.

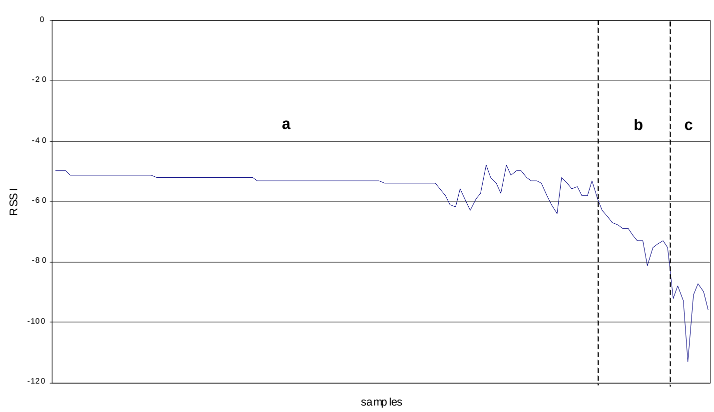


Figure 6.17: RSSI measurement driving on route B

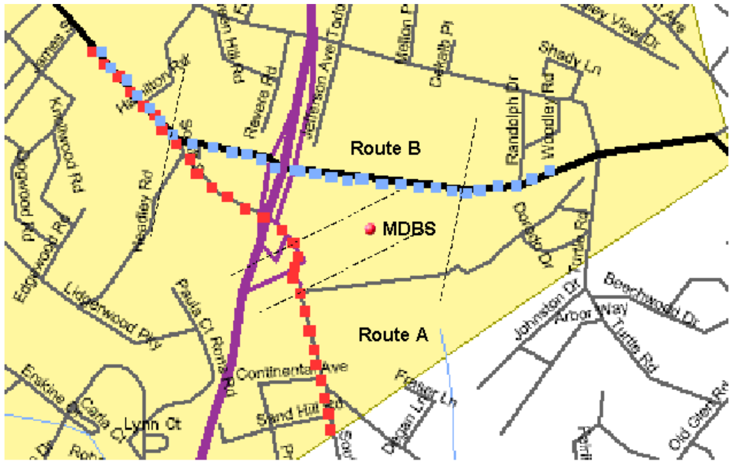


Figure 6.18: Building an RSSI interval map.

Driving on route A, we obtain the values of RSSI [1] showed in Figure 6.16. The peak of the RSSI (i.e. the lowest RSSI in dB), approximately corresponds to the closest location to the cell tower. We can identify at least three RSSI intervals (a:[-40db,-60dB],b:[-60dB,-80dB],c:[-80dB, -120dB]). Using this map we can assert that if a user is connected to the base station MDBS, and measures an RSSI lower than $-60dB$ (region a), then he/her is reasonably close to the MDBS, on the contrary if the RSSI is in the region c, then he/her is fairly distant from the MDBS. Driving on route B we obtain the values of RSSI showed in Figure 6.17. In this case is not possible to identify an RSSI peak, and furthermore the figure shows an almost constant behavior for the first half of the samples (region a). So the improvements in accuracy that can be achieved through RSSI measurements are less accurate than in the previous case (the region a is too large). The experiment in Figure 6.19 shows the RSSI map while moving along the NJ transit railway from Hudson Essex (the train left from NY Penn Station) to Maplewood. The regions a,b,c,d, correspond to points in which the MT is connected to a specific cell. Observe that the RSSI behavior is almost a stair function in proximity of the cell region while it is more jagged in the boundary region except where no connection is available ($-113dB$). This behavior allows us to better localize a user. If we consider region c, we can easily define two further regions, approximately corresponding to values of RSSI of $-77dB$ and $-94dB$. Thus for example, if we know a commuter being connected to the cell tower covering region c, and the RSSI being close to one of the above values, we can easily assert whether the train is closer to

[1] The values of the RSSI can be easily gathered through the MSCI protocol, using the Status Request function

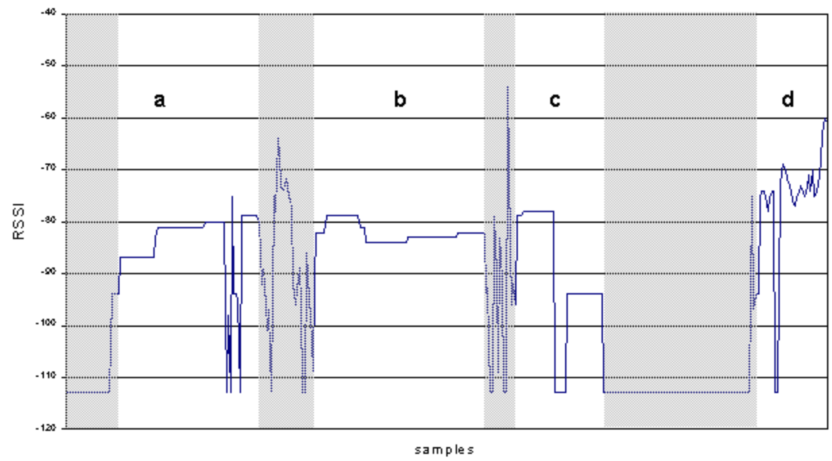


Figure 6.19: RSSI measurements on the railroad.

region b or to region d. Furthermore, between the $-77dB$ region and $-94dB$ region, there is a narrow discontinuity ($RSSI = -113dB$) that can be used to detect when the train transits from one region to the other.

6.7 Conclusions and Future Works

In this chapter we have presented a system architecture whose purpose is extracting location information of CDPD subscribers from a wireless device and making it available to a location server. Moreover we have discussed the accuracy of this location technique and we have implemented some other techniques to improve such accuracy. The main objective of the future work will be the implementation of a similar solution in the GSM environment. We are implementing an extension of the PPP protocol that allows us to interact with any modem (not being restricted to MSCI enabled modem). We are also exploring the location potentialities offered by the SIMToolkit. The SIMToolkit is a tool to program the mobile SIM phone in order to provide new customized services.

Bibliography

[1] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive online routing. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 32–40, 1993.

[2] B. Awerbuch, R. Gawlick, T. Leighton, and Y. Rabani. On-line admission control and circuit routing for high performance computing and communication. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 412–423, 1994.

[3] K.R. Baker. Introduction to sequencing and scheduling. *Wiley*, 1975.

[4] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, and Randy H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, 1997.

[5] Nicholas Bambos. Toward power-sensitive network architectures in wireless communications: concepts, issues, and design aspects. *IEEE Personal Communications*, 5(3):50–59, June 1998.

[6] Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, Joseph (Seffi) Naor, and Baruch Schieber. A unified approach to approximating resource allocation and scheduling. In *Proceedings of the Thirty-Second Annual ACM Symposium on the Theory of Computing*, Las Vegas, Nevada, 2000.

[7] Amotz Bar-Noy, Ran Canetti, Shay Kutten, Yishay Mansour, and Baruch Schieber. Bandwidth allocation with preemption. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, pages 616–625, Las Vegas, Nevada, 29 May–1 June 1995.

[8] Amotz Bar-Noy, Sudipto Guha, Joseph (Seffi) Naor, and Baruch Schieber. Approximating the throughput of multiple machines under real-time scheduling. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 622–631, Atlanta, GA USA, 1–4 May 1999.

[9] Y. Bartal, A. Fiat, and S. Leonardi. Lower bounds for on-line graph problems with application to on-line circuit and optical routing. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 531–540, 1996.

[10] Y. Bartal and S. Leonardi. On-line routing in all-optical networks. In *Proceedings of the 24th International Colloqium on Automata, Languages and Programming*, LNCS 1256, pages 516–526. Springer-Verlag, 1997.

[11] S. Ben-David, A. Borodin, R.M. Karp, G. Tardos, and A. Widgerson. On the power of randomization in on-line algorithms. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, 1990.

[12] Paul Bender, Peter Black, Matthew Grob, Roberto Padovani, Nagabhushana Sindhushayana, and Andrew Viterbi. CDMA/HDR; A bandwith-efficient high-speed wireless data service for nomadic users. *IEEE Communications Magazine*, 38(7):70–77, July 2000.

[13] Y. Bernet. A framework for integrated services operation over diffserv networks. *Internet Draft*, May 2000.

[14] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Transactions on Communications* , 44(10):1261 –1271, October 1996.

[15] G. Bianch and N. Blefari-Melazzi. Grip: Qos support over stateless diffserv networks by means of localized measurements and decisions. *Lecture Notes on Computer Science, Springer-Verlag*, 1989, 1989.

[16] G. Bianchi and N. Blefari-Melazzi. A migration path to provide end-to-end qos over stateless networks by means of a probing-driven admission control. *Internet Draft*, March 2001.

[17] S. Blade, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. *RFC2475*, December 1998.

[18] Helmut Bolcskei, A. Paulraj, K V S. Hari, Rohit U Nabar, and Willie W. Lu. Fixed Broadband Wireless Access: State of the Art, Challenges and Future Directions. *IEEE Communications Magazine*, 39(1):100–108, January 2001.

[19] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[20] R. Braden, L Zhang, S. Berson, S. Herzog, and S. Jamin. Resource re servation protocol (rsvp) - version 1 functional specification. *RFC2205*, September 1997.

[21] L. Breslau, S. Jamin, and S. Schenker. Comments on the performance of measurement-based admission control algorithms. *IEEE INFOCOM 2000*, March 2000.

[22] L. Breslau, E. W. Knightly, S. Schenker, I. Stoica, and H. Zhang. Endpoint admission control: Architectural issues and performance. *ACM SIGCOMM 2000*, August 2000.

[23] A. De Carolis. Qos-aware handover support in mobile ip: Secondary home agents. *IETF-DRAFT.*

[24] Justin Chuang and Nelson Sollenberger. Beyond 3G: Wideband data access based on OFDM and dynamic packet assignment. *IEEE Communications Magazine*, 38(7):78–87, July 2000.

[25] J M. Cioffi, G P. Dudevoir, M V. Eyuboglu, and G D. Forney. MMSE decision-feedback equalizers and coding: Parts I & II. *IEEE Transactions on Communications*, 43(10):2582–2604, Oct. 1995.

[26] P. Conforto, C. Tocci, G. Losquadro, F. Fedi, R.E. Sheriff, and A. Vitaletti. Suited/gmbs system: Architecture and mobile terminal. *IST Mobile Communication Summit 2001*, September 2001.

[27] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* John Wiley and Sons, Inc., New York, 1991.

[28] J.Wein D.Karger, C.Stein. Scheduling algorithms. *Algorithms and Theory of Computation Handbook, CRC Press*, 1999.

[29] http://www.estec.esa.nl/artes3/projects/12telbios/telbios.htm.

[30] CDPD Forum. Cellular digital packet data system specification, release 1.1. January 1995.

[31] Juan A. Garay, Inder S. Gopal, and Shay Kutten. Efficient on-line call control algorithms. *Journal of Algorithms*, 23(1):180–194, April 1997.

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

[33] Michael R. Garey and David S. Johnson. *Computers and intractability : a guide to the theory of NP-completeness.* W H. Freeman., San Francisco, 1979.

[34] Jordan Gergov. Approximation algorithms for dynamic storage allocation. In *European Symposium on Algorithms (ESA'96)*, volume 1136 of *Lecture Notes in Computer Science*, pages 52–61. Springer, 1996.

[35] Jordan Gergov. Algorithms for compile-time memory optimization. In *Proc. of the 10th ACM-SIAM Symposium on Discrete Algorithms*, pages 907–908, 1999.

[36] O. Gerstel, G.H. Sasaki, and R. Ramaswami. Dynamic channel assignment for WDM optical networks with little or no wavelength conversion. In *Proceedings of the 34th Allerton Conference on Communication, Control, and Computing*, 1996.

[37] M. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, NY, 1980.

[38] M. Grossglauser and D. N. C. Tse. A time-scale decomposition approach to measurement-based admission control. *IEEE INFOCOM 1999*, March 1999.

[39] M.M. Halldórsson and M. Szegedi. Lower bounds for on-line graph coloring. In *Proceedings of the 2nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 211–216, 1992.

[40] S. Irani. Coloring inductive graphs on-line. In *Proceedings of the of 31th Annual IEEE Symposium on Foudations of Computer Science*, pages 470–479, 1990.

[41] http://www.ist-suited.com/.

[42] General aspects of quality of service and network performance in digital networks, including isdns. *ITU-T Recomm. I.350*, March 93.

[43] G Borriello J. Hightower. Location systems for ubiquitos computing. *Computer, IEEE Computer Society*, 34(8):57–66, August 2001.

[44] Niranjan Joshi, Srinivas R. Kadaba, Sarvar Patel, and Ganapathy Sundaram. Downlink scheduling in CDMA data networks. In *ACM MobiCom*, pages 179–190, 2000.

[45] B. Kalyanasundaram and K.Pruhs. Speed is as powerful as clairvoyance. In *IEEE Symposium on Foundations of Computer Sci.*, pages 214–221, 1995.

[46] H. A. Kierstead. A polynomial time approximation algorithm for dynamic storage allocation. *Disccrete Mathematics*, 88:231–237, 1991.

[47] H. A. Kierstead and W. T. Trotter. An extremal problem in recursive combinatorics. *Congressus Numerantium*, 33:143–153, 1981.

[48] H. A. Kierstead and W. T. Trotter. On-line graph coloring. In Lyle A. McGeoch and Daniel D. Sleator, editors, *On-line Algorithms*, volume 7 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 85–92. AMS/ACM, February 1991.

[49] D.E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 2nd Edition.* Addison-Wesley, 1973.

[50] Jean-Marc Latapy. *Gsm mobile station locating.* PhD thesis, Norwegian University of Science and Technology, 1996.

[51] L.Becchetti, S.Diggavi, S.Leonardi, A. Marchetti-Spaccamela, S. Muthukrishnan, T. Nandagopal, and A. Vitaletti. Scheduling algorithms for next generation multirate wireless networks. *AT&T Labs Research Report*, Jan 2001.

[52] William C. Y. Lee. *Mobile Communications Engineering.* McGraw-Hill Telecommunications, 2 edition, 1998.

[53] S. Leonardi, A. Marchetti-Spaccamela, A. Presciutti, and A. Rosèn. On-line randomized call-control revisited. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, pages 323–332, 1998.

[54] S. Leonardi and A. Vitaletti. Randomized lower bounds for online path coloring. *RANDOM 98: International Workshop on Randomization and Approximation Techniques in Computer Science, Barcelona, Spain*, pages 232–247, October 1998. Preliminarly accepted to Information and Computation.

[55] L. Lovász, M. Saks, and W.T. Trotter. An on-line graph coloring algorithm with sublinear performance ratio. *Discrete Mathematics*, 75:319–325, 1989.

[56] Durlacher Research Ltd. Mobile commerce report, 1998.

[57] Yuming Lu and R W. Brodersen. Integrating power control, error correction coding, and scheduling for a CDMA downlink system. *IEEE Selected Areas in Communications*, 17(5):978–989, May 1999.

[58] David G. Luenberger. *Linear and Non-linear programming.* Addison-Wesley, Reading, Mass., 2nd edition, 1984.

[59] V. Marziale and A. Vitaletti. A framework for internet qos requirements definition and evaluation: an experimental approach. *IST Mobile Communication Summit 2001*, September 2001.

[60] S.Muthukrishnan M.Bender, S.Chakrabarti. Flow and stretch metrics for scheduling continuous job streams. *Proceedings of Annual Symposium on Discrete Algorithms (SODA '98)*, pages 270–279, 1998.

[61] C.Murta M.Crovella, M.Harchol-Balter. *Task Assignment in a Distributed System: Improving Performance by Unbalancing Load. Technical Report 97-018*, Boston University, 1997.

[62] M.Harchol-Balter M.Crovella, R.Frangioso. *Connection Scheduling in Web Servers. Technical Report 99-003*, Boston University, 1999.

[63] Sanjiv Nanda, Krishna Balachandran, and Sarath Kumar. Adaptation techniques in wireless packet data services. *IEEE Communications Magazine*, 38(1):54–65, January 2000.

[64] K. Nichols, S. Blake, F. Baker, and D. Black. Definitions of the differentiated service field (ds field) in the ipv4 and ipv6 headers. *RFC2474*, December 1998.

[65] E. Perkins. Ip mobility support. *RFC2002*, October 1996.

[66] E. Perkins. Mobile ip. *Communications Magazine*, pages 84–99, May 1997.

[67] Ampl: A modeling language for mathematical programming. http://www.ampl.com.

[68] Loqo optimization toolkit, 2000. http://www.orfe.princeton.edu/ loqo.

[69] S. Muthukrishnan R. Jana, T. Johnson and A. Vitaletti. Location based services in a wireless wan using cellular digital packet data (cdpd). *MOBIDE 2001: Second ACM international workshop on Data engineering for wireless and mobile access, Santa Barbara, CA USA*, May 2001.

[70] P. Raghavan and E. Upfal. Efficient routing in all-optical networks. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 133–143, 1994.

[71] Theodore S. Rappaport. *Wireless Communications, Principles & Practice*. Prentice Hall, Inc., Piscataway, New Jersey, 1996.

[72] R.Jana, T.Johnson, S.Muthukrishnan, and A. Vitaletti. System and method for enabling data-mode location aware mobile services. United States provisional patent application.

[73] R.Jana, T.Johnson, S.Muthukrishnan, and A. Vitaletti. System and method for enabling handset-based network enhanced location services. United States provisional patent application.

[74] R.Jana, T.Johnson, S.Muthukrishnan, and A. Vitaletti. System and method for handset-assisted, location tracking with multiple servers and services. United States provisional patent application.

[75] A. Marchetti-Spaccamela S. Leonardi and A. Vitaletti. Approximation algorithms for bandwidth and storage allocation problems under real time constraints. *FST TCS 2000: Foundations of Software Technology and Theoretical Computer Science 20th Conference, New Delhi, India*, pages 409–420, December 2000. Submitted to Journal of Algorithms.

[76] S.Muthukrishnan S.Acharya. Scheduling on-demand broadcasts for heterogenous: new metrics and algorithms. *ACM Mobicom*, pages 43–54, 1998.

[77] Prasun Sinha, Narayanan Venkitaraman, Raghupathy Sivakumar, , and Vaduvur Bharghavan. WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks. In *ACM MobiCom*, pages 231–241, 1999.

[78] D. Sleator and R.E. Tarjan. Amortized Efficiency of List Update and Paging Rules. *Communications of ACM*, 28:202–208, 1985.

[79] D. Sleator and R.E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of ACM*, 28:202–208, 1985.

[80] M. Slusarek. A coloring algorithm for interval graphs. In *Proc. of the 14th Mathematical Foundations of Computer Science*, pages 471–480, 1989.

[81] F.C.R. Spieksma. On the approximability of an interval scheduling problem. *Journal of Scheduling*, 2:215–2227, 1999.

[82] Thomas J. J. Starr and John Cioffi. *Understanding digital subscriber line technology.* Prentice Hall, Inc., Upper Saddle River, New Jersey, 1999.

[83] www.cs.washington.edu/research/portolano/papers/UW-CSE-01-07-01.pdf.

[84] V.Bharghavan T.Nandagopal, S.Lu. A unified architecture for the design and evaluation of wireless fair queueing algorithms. *ACM Mobicom*, pages 132–142, 1999.

[85] S. Vishwanathan. Randomized on-line graph coloring. *Journal of Algorithms*, 13:657–669, 1992.

[86] Andrew J. Viterbi. *CDMA : principles of spread spectrum communication.* Addison-Wesley Pub. Co., Reading, Mass., 1995.

[87] W.Almesberger, T.Ferrari, and J. Y. Le Boudec. Srp: a scalable resource reservation protocol for the internet. *IWQoS'98*, May 1998.

[88] Ellen Kayata Wesel. *Wireless multimedia communications : networking video, voice, and data.* Addison-Wesley, Inc., Reading, Mass., 1998.

[89] P. P. White. Rsvp and integrated services in the internet: a tutorial. *IEEE Communications magazine*, 35(5), May 1997.

[90] Svein Yngvar Willassen. *A method for implementing Mobile Station Location in GSM.* PhD thesis, Norwegian University of Science and Technology, 1998.

[91] NOVATELL WIRELESS. Notes on the modem status / configuration interface protocol, v1.8, March 2000. http://www.novatelwireless.com/support/download/MSCI_Protocol.pdf.

[92] J. Wroclawski and A. Charny. Integrated service mapping for differentiated services networks. *Internet Draft*, March 2000.

[93] J: Wroclawsky. The use of rsvp with ietf integrated services. *RFC2210*, September 1997.

[94] A. Yao. Probabilistic computations: Towards a unified measure of complexity. In *Proceedings of the 17th Annual IEEE Symposium on Foundations of Computer Science*, pages 222–227, 1977.