INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Adaptive Quality of Service Management in Wireless/Mobile Networks

By

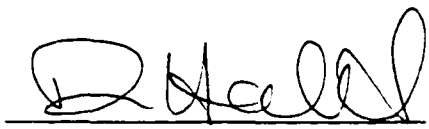
Mahmoud R Sherif

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

The City University of New York

2000

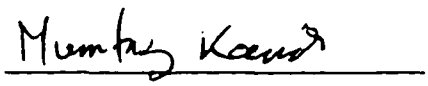
This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.



Prof. Ibrahim Habib

Date 3-10-2000

Chair of Examining Committee



Dean. Mumtaz Kassir

Date 4-11-2000

Executive officer

Prof. Tarek Saadawi (EE Dept. CCNY,CUNY)

Prof. Mohamed Ali (EE Dept. CCNY,CUNY)

Prof. Myung Lee (EE Dept. CCNY,CUNY)

Dr. Mahmoud Naghshineh (IBM T.J. Watson)

(external examiner)

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

Adaptive Quality of Service Management in Wireless/Mobile Networks

By

Mahmoud R Sherif

Adviser: Professor Ibrahim Habib

In wireless ATM based networks, admission control is required to reserve resources in advance for calls requiring guaranteed services. In the case of a multimedia call, each of its substreams (i.e. video, audio and data) has its own distinct quality of service (QoS) requirements (e.g. cell loss rate, delay, jitter, etc.). The network attempts to deliver the required QoS by allocating an appropriate amount of resources (e.g., bandwidth, buffers). The negotiated QoS requirements constitute a certain QoS level that remains fixed during the call (static allocation approach). Accordingly, the corresponding allocated resources also remain unchanged. In this thesis, an adaptive allocation of resources algorithm based on genetic algorithms is proposed and analyzed. In contrast to the static approach, each substream declares a preset range of acceptable QoS levels (e.g., high, medium, low) instead of just a single one. As the availability of resources in the wireless network varies, the algorithm selects the best possible QoS level that each substream can obtain. In case of congestion, the algorithm attempts to free up some resources by degrading the QoS levels of the existing calls to lesser ones. This is done, however, under the constraint of achieving maximum utilization of the resources while, simultaneously, distributing them fairly among the calls. The degradation is limited to a minimum value predefined in a user defined profile

(UDP). Genetic algorithms have been used to solve the optimization problem. From the user perspective, the perception of the QoS degradation is very graceful and happens only during over-load periods. The network services, on the other hand, are greatly enhanced due to the fact that the call blocking probability is significantly decreased. Simulation results, for a cellular-based wireless network, demonstrate that the proposed algorithm performs well in terms of increasing the number of admitted calls while utilizing the available bandwidth fairly and effectively. The thesis also proposes a QoS based channel borrowing algorithm (QCBA) that allows a cell (in a cellular system) to borrow some free channels from any neighboring cell to accommodate new calls. The criteria for choosing the free channel include not only the number of free channels but also the QoS levels in the donor cell. The criteria is also extended to include the effect of channel locking on the number of free channels and the QoS levels on the locked cells. Simulation results shows that the QCBA decreases the call blocking rate while increasing the average QoS level.

Acknowledgements

I would like first to express sincere thanks to my supervisor Professor Ibrahim Habib who has provided me with invaluable academic advice. He devoted much of his time to discussing details of this research with me and helping me in so many other ways. I have been greatly influenced by his methodology in the field of scientific research, and have benefited from his detailed comments on every aspect related the research in this thesis.

I would also like to thank the Electrical Engineering faculty and specially the examining committee for their support and for giving me the chance to achieve my goals.

Special thanks to Dr. Mahmoud Naghshineh for his helpful support.

Finally, I would like to thank my family for their support all the time. They were always there and gave me the love and support when I needed them the most. Very special thanks for my wife who helped me a lot in so many ways. Without her, this work would not have been possible.

This dissertation is dedicated to my loving parents and wife.

Contents

Chapter 1: Introduction 1

1.1 Wireless Mobile ATM 1

1.1.1 Mobile Multimedia Networks Architecture 2

1.2 QoS Provisioning in Wireless/Mobile Networks 4

1.2.1 Non-adaptive versus Adaptive QoS Provisioning 6

1.2.2 Adaptive QoS provisioning 7

1.2.2.1 Dynamic QoS Management 8

1.2.2.2 Adaptive End-to-end QoS Provisioning Framework 10

1.2.2.3 Terminal QoS for Adaptive Applications 13

1.2.2.4 QoS-Aware Resource Management Scheme (QualMan) 15

1.2.2.5 Peer-to-peer QoS Management 19

1.2.2.6 QoS Management for News-on-demand Applications 21

1.2.2.7 Summary of Adaptive QoS Provisioning Schemes 22

1.3 Proposed Approach 24

1.4 Adaptive Allocation of Resources Algorithm 25

1.5 QoS Levels for Multimedia Substreams 28

1.6 Genetic Algorithms 31

Chapter 2: Adaptive Allocation of Resources Algorithm 42

2.1 Genetic Algorithm Module I 48

2.2 Genetic Algorithm Module II 49

Chapter 3: Simulation Results..... 57

3.1 Case Study 1..... 58

3.2 Case Study 2..... 64

Chapter 4: A QoS based Channel Borrowing Algorithm..... 83

4.1 Background 84

4.2 Related Work..... 85

4.3 Suggested Channel Borrowing Algorithm..... 86

4.3.1 Module III..... 91

4.4 Simulation Results..... 97

Chapter 5: Conclusions and Discussions..... 110

References..... 112

Publications..... 117

List of Figures

Fig. 1.1 Mobile Networks Architecture..... 4

Fig. 1.2 Simplified High-level diagram of the DQM..... 9

Fig. 1.3 Layers of a multiresolution stream..... 11

Fig. 1.4 Application that adapt to packet loss feedback from the network..... 15

Fig. 1.5 QoS Broker during Connection Setup..... 17

Fig. 1.6 Scaling Filter Example..... 20

Fig. 1.7 Entities in a QoS Management System..... 25

Fig. 1.8 Database used by the algorithm..... 27

Fig. 1.9 Example of a User-Defined Profile (UDP)..... 27

Fig. 1.10 Example of QoS levels for video, audio and data..... 28

Fig. 1.11 Live MPEG captured frames..... 31

Fig. 1.12 Example of crossover operation..... 37

Fig. 1.13 Example of mutation operation..... 38

Fig. 2.1 Required Bandwidth versus QoS indices..... 45

Fig. 2.2 Block Diagram of the Proposed Allocation of Resources Algorithm... 46

Fig. 2.3 Genetic Algorithm Module I..... 50

Fig. 2.4 Decomposition Algorithm..... 54

Fig. 2.5 Genetic Algorithm Module II..... 55

Fig. 3.1 First System Under Study..... 59

Fig. 3.2 Bandwidth Requirement versus Time..... 59

Fig. 3.3 Number of calls versus time..... 60

Fig. 3.4 Average QoS index versus Time..... 61

Fig. 3.5 Blocking Rate versus Time..... 62

Fig. 3.6 QoS Histogram..... 63

Fig. 3.7 Second System Under Study..... 64

Fig. 3.8 Bandwidth Requirement versus time..... 66

Fig. 3.9 Number of calls versus time in cell 1..... 67

Fig. 3.10 Number of calls versus time in cell 2..... 68

Fig. 3.11 Average QoS index versus Time for cell 1..... 69

Fig. 3.12 Average QoS index versus Time for cell 2..... 70

Fig. 3.13 Blocking Rate versus Time for cell 1..... 71

Fig. 3.14 Snapshot of the simulator output trace file for cell 1..... 72

Fig. 3.15 Link Utilization versus Number of Calls..... 73

Fig. 3.16 Handoff from cell 2 to cell 1..... 74

Fig. 3.17 Handoff from Cell 2 to Cell 1..... 75

Fig. 3.18 Percentage gain versus number of calls for Cav=25Mbps..... 77

Fig. 3.19 Percentage gain versus number of calls for Cav=75Mbps..... 78

Fig. 3.20 Percentage gain versus number of calls for Cav=155Mbps..... 79

Fig. 3.21 Percentage gain versus available capacity..... 80

Fig. 3.22 No. of calls versus time..... 81

Fig. 4.1 (a) Cell cluster (N=7), (b) A seven-cell reuse plan..... 85

Fig. 4.2 Block Diagram of the Algorithm (after including channel borrowing). 88

Fig. 4.3 Genetic Algorithm Module I..... 89

Fig. 4.4 Module Chooser..... 90

Fig. 4.5 Two Tier (N=7) Cell Layout..... **91**

Fig. 4.6 Flow of messages between acceptor, candidate and affected cells..... **96**

Fig. 4.7 System under study..... **98**

Fig. 4.8 Initial call distribution..... **100**

Fig. 4.9 Call blocking versus offered load..... **103**

Fig. 4.10 Reduction in Call blocking versus offered load..... **104**

Fig. 4.11 Increase in offered load versus call blocking **105**

Fig. 4.12 QoS enhancement versus offered load..... **106**

Fig. 4.13 QoS enhancement versus time..... **108**

List of Tables

Table 1.1 Summary of Adaptive QoS Provisioning Schemes..... **23**

Table 2.1 Translation table for the QoS indices..... **44**

Table 4.1 Module III simulator information (initial call distribution) **102**

Table 4.2 Module III simulator information (higher traffic load)..... **102**

Chapter 1: Introduction

Asynchronous Transfer Mode (ATM) is the transfer mode of choice for Broadband Integrated Services Digital Network (B-ISDN) chosen by the ITU-T [1]. The ATM technique provides an attractive solution for integrating different types of services. It is designed to provide multimedia traffic services (e.g., video, audio, data). These services are likely to have a wide range of both traffic characteristics, performance, and quality of service (QoS) requirements.

Extending broadband multimedia services to mobile wireless terminals is the next step in the telecommunications industry. Wireless mobile ATM (wmATM) technologies provide innovative solutions to constructing a generic broadband wireless core to ensure wireless quality of service and mobility management. This wmATM core also acts as a wideband open platform to support different air interfaces and multidimensional wireless systems. The packetized air link helps dynamic bandwidth allocation, and therefore improves wireless spectrum utilization.

1.1 Wireless Mobile ATM

WmATM evolves from wireless ATM, and is supported with a direct signaling and packet-based mobile control architecture which can be attractively applied in next-generation broadband wireless systems, including broadband wireless mobile and broadband wireless access networks.

The ITU-IMT-2000 [2] third-generation (3G) wireless communication system is the major player for next-generation broadband wireless mobile in the global communications market.

Many 3G wireless study committees are taking the initiative to produce their own proposals for radio transmission technologies and advanced system prototypes for IMT-2000. The wmATM enhanced 3G wireless system provides a good solution to implement the harmonized wireless communications required by the IMT-2000 [2].

1.1.1 Mobile Multimedia Networks Architecture

The mobile part of future network architectures can be viewed as consisting of a set of overlapping tiers, each with its own specific characteristics. Satellite, micro, and pico-cellular segments will each cover widely varying geographic areas and support different data rates for mobile terminals within them. Some tiers will be privately operated, others publicly operated. As shown in Fig.1.1, pico cells, with data rates in excess of 25 Mb/s, will typically cover building scale areas; micro cells, with up to 2 Mb/s rates, will cover dense urban areas; macro cells, with several 100s of Kb/s, will provide wide-area coverage; and satellite segments, giving per terminal data rates from 144 Kb/s or more, will cover up to entire continents.

Future mobile multimedia systems will serve 3G terminals in any combination of these environments. Since adaptive radio technology for such a range of transmission schemes (modulation schemes as well as data rates) will not be available for some time, it is reasonable to expect that initial 3G mobile multimedia terminals will incorporate multiple radio interfaces. For example, 3G 8/1 of ITU-R is considering harmonization of Radio Transmission Technique proposal to become the standard for the radio interface(s) of ITU IMT-2000 [2]. While this process is still ongoing at the time of this writing, it can be anticipated that there will definitely be different radio interfaces for terrestrial- and satellite-based mobile systems.

Traditionally, devices used for communication are connected to one type of network. The universal interconnection of land-based communications networks (here we mean cable, fiber, and copper networks forming the vast majority of current telecommunications networks and the Internet) provides global communications links to any fixed system. Gateways from mobile networks, such as the Global System for Mobile Communication (GSM), to the land-based networks extend this global connectivity to mobile terminals. Specific components of the mobile network deal with terminal identification and authentication for registration with the network and management of the terminal's location as it moves, and other aspects of the network manage the rerouting of traffic as the terminal passes from one coverage area to another during the course of the same connection or session [3]. The rerouting mechanisms are called handoff mechanisms and are a key part of mobile network architecture [4]. Some systems incorporate multiple network interfaces. This idea has been carried forward into the universal mobile Telecommunications system (UMTS). The current proposal uses GSM to provide wide-area coverage supplemented with Wideband code Division Multiple access (WCDMA) micro and macro cells to provide higher bandwidth areas. The proposed network intelligence in UMTS will then provide the ability to handover between the GSM and WCDMA layers.

Fast lossless handover between different network types is one of the greatest challenges in building seamless mobile multimedia networks. It can be readily seen from the tiered model of 3G systems that the solution to this problem will be key to successful integration of multimedia.

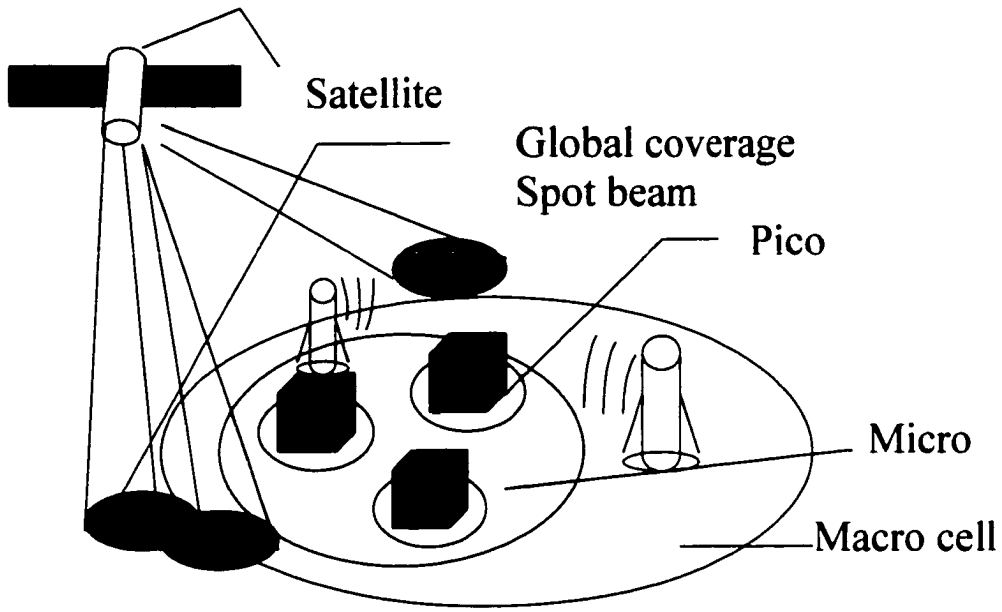


Fig 1.1 Mobile Networks Architecture.

1.2 QoS Provisioning in Wireless/Mobile Networks

Compared to fixed networks, the fluctuation in resource availability in wireless and mobile networks is much more severe and is mainly caused by two basic characteristics of such networks: (1) fading and (2) mobility [3]-[6]. First of all, in contrast to links of fixed networks, wireless links suffer from a high bit error rate (BER) and fading. This results in packet loss which is then translated into packet delay and jitter. Using highly sophisticated hardware/software functions at the receiver and/or transmitter can improve the level of BER. However there is a limit for such improvement. Furthermore, the use of these sophisticated functions might cause more packet delays. Fading in a wireless channel is highly varying with time and spatial dependencies. Adding interference to this, we get a transmission link

with a highly varying bandwidth. The second main reason for the fluctuation in availability of network resources is mobility and handoff. As a mobile station roams in the wireless network and hands off from one access point to another, there is a change in the wireless resources (i.e., the radio cell corresponding to the new access point) and wired resources (i.e., links & switches in the fixed network that constitute a path to a new, base station) [6]. This change in network resources can result in a major fluctuation in the availability of network resources reserved for the connection. Also, depending on the rerouting protocol and the signaling system overhead, the level of QoS degradation due to the connection outage during a handoff can be different. QoS provisioning for wireless/mobile networks has to address the high level of fluctuation in the availability of network resources as described above. The existing QoS components and functions designed for fixed (or wired) networks need to be enhanced to provide multimedia mobile and wireless services. Considering the explosion of World Wide Web and multimedia-based applications as well as enormous recent popularity of mobile and handheld devices capable of wireless communication, it is becoming a major and essential issue to provide networking architectures to address these issues. Hence, there is a need for comprehensive QoS disciplines that can ensure per-connection, end-to-end performance guarantees [7]-[9].

QoS provision is comprised of the following components: (1) QoS mapping, (2) admission testing, and (3) resource allocation, see [7]. The QoS mapping translates between representations of QoS at different system levels. For example, the transport level QoS specification may be expressed in terms of average and peak bit rate, jitter, and delay constraints. For admission testing and resource allocation, this representation must be translated to something more meaningful (e.g., in terms of some user-defined QoS levels).

Admission testing is responsible for comparing the resource requirement arising from the requested QoS against the available resources in the system. Resource allocation, on the other hand, arranges for the allocation of suitable network resources according to the user QoS specifications.

QoS provisioning for multimedia applications in wireless/mobile networks needs some form of adaptation to be performed on the multimedia substreams. In other words, the multimedia applications need to adapt to changes in the availability of network resources. The motives of such adaptation requirement can be demonstrated by the following simple example. For a video connection, the quality of video received depends heavily on packet loss and/or jitter due to fading and handoff. Therefore, real-time applications that require more QoS guarantees are more susceptible to the QoS fluctuations. Eventually, the QoS guarantees of the video connection will not be met.

1.2.1 Non-adaptive versus Adaptive QoS Provisioning

The solution to the presented QoS provisioning problem in a wireless/mobile environment can be categorized into two categories: non-adaptive approach and an adaptive approach. In the non-adaptive approach, the wireless/mobile network is required to match the QoS guarantees of a fixed network. That is: despite major fluctuations in the availability of network resources throughout the connection time, the network should maintain the QoS level promised at connection setup: The advantage of this solution is that it does not require a change in the existing application-to-network service specification. However, there is a serious concern with this approach. To maintain a fixed service guarantee, there are two general approaches. In the first approach the network must resort to providing a fixed level of

QoS (e.g., that provided, by cellular telephone networks today) which can be guaranteed in the presence of fluctuations in wireless resource availability. This approach does not support a wide range of future wireless multimedia applications. In another approach, the service providers have to increase the degree of preprovisioning in the network to a very high degree to mitigate packet delay and loss due to handoff and fading in wireless links under all conditions. The result of this overprovisioning is that support of a wide range of integrated services over wireless and mobile networks becomes economically impossible.

An alternative approach would be an adaptive QoS provisioning approach where the network and the application share the responsibility of providing the required QoS and the QoS guarantees simultaneously. The next section will describe this approach and provide a survey of the research done in this area.

1.2.2 Adaptive QoS provisioning

In the adaptive approach, the end-to-end QoS provisioning is no longer the sole responsibility of the network or the application. Instead they both share the responsibility to deliver the multimedia content in the most acceptable way. The adaptive QoS provisioning is therefore comprised of two main components: The network-based provisioning responsibility and the application-based responsibility. A number of papers dealt with this adaptive approach and each presented its own way of defining responsibilities among the network and the application. In the next sections, we are going to describe a number of these papers.

1.2.2.1 Dynamic QoS Management

In [10], a dynamic QoS management (DQM) scheme for the control and management of multilayer coded flows operating in heterogeneous multimedia networking environment is presented. The DQM describes functions such as video and audio source coding, resource reservation, flow scheduling, and QoS filtering and adaptation. The DQM proposes the use of two techniques, the dynamic rate shaping and the adaptation network service. These directly correspond to the application-based responsibility and the network-based responsibility. The dynamic rate shaping is a source-based (application responsibility) QoS filter for manipulating the rate of MPEG-coded flows, matching it to the available network resources while minimizing the distortion observed at the receiver. In the adaptive network service (network-based responsibility) the multilayer coded flows is divided into a base layer in addition to a group of enhancement layers. The adaptive network service defines a set of hard guarantees to the base layer, and fairness guarantees to the enhancement layers based on a new bandwidth allocation technique called "weighted fair allocation". The primary focus of this paper was the MPEG-2 coded video. Within this framework, two alternatives were suggested to achieve the QoS adaptation for MPEG-2 coded video: intrinsic and extrinsic techniques. The intrinsic adaptation technique was presented in the form of scalability profiles. Four scalability modes are considered: spatial, SNR, temporal and data partitioning. Detailed examples of the use and implementation of these scalable modes are also presented in the paper. The extrinsic adaptation technique, in the form of an adaptive service, is performed by QoS filters that operate directly on compressed streams, performing the desired manipulations. The goal of the adaptive service is to admit as many base layers as possible to meet the hard guarantees.

In general, this paper presents a detailed adaptive QoS provisioning scheme. However, the paper does not present the details of the QoS mapping component or the admission testing component of the QoS provisioning system. Fig. 1.2 shows a simplified high-level diagram of the DQM.

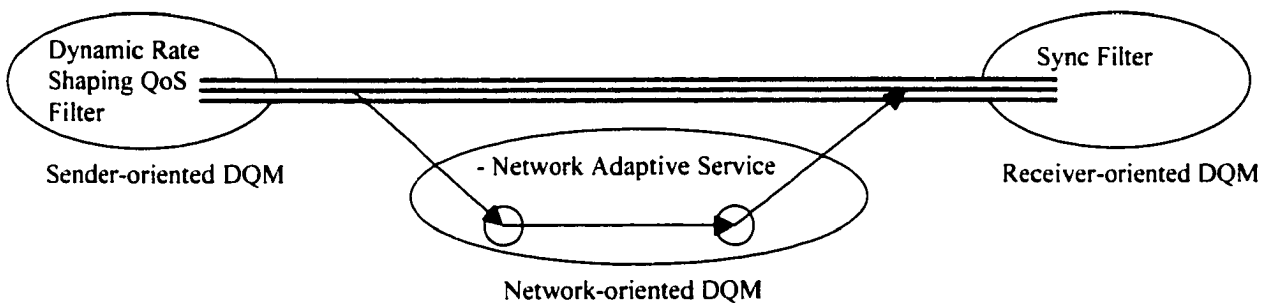


Fig. 1.2 Simplified High-level diagram of the DQM

In [11], the DQM is extended to be applied on a wireless multimedia system. Wireless media scaling is also presented in details showing how the DQM can be applied to wireless/mobile environment. A QoS-aware mobile middleware capable of automatically scaling flows during handoff and during other periods of persistent QoS fluctuations is presented. The notion of having different QoS levels for each of the multimedia substreams is described. The notion of user-defined profiles is also introduced in addition to a general framework for QoS management. The main contribution of the paper is the presentation of the mobile middleware, called "mobiware", in terms of design, implementation and performance issues. The scheme however does not provide any details in regards to the real time application of such scheme and the performance issues in such a case. Furthermore, the admission control and the allocation of resources are out of the scope of the paper.

1.2.2.2 Adaptive End-to-end QoS Provisioning Framework

An adaptive end-to-end QoS provisioning framework for wireless and mobile networks has been proposed in [5]. In this framework, multimedia applications are required to be able to accept varying degrees of network guarantee levels. This has been referred to as "multiresolution" applications. In this context, multimedia applications need to be adaptive and network-aware, and networks must provide application-aware services. The paper presents different available techniques of multimedia stream adaptation. The requirements and the required components of an adaptive mobile and wireless network are also described. The application-based responsibility mainly lies in the ability to provide the network with a multiresolution stream. This multiresolution stream consists of three substreams (video, audio and data) each consisting of base (A,B,C) and enhancement layers + and ++. Fig. 1.3 shows such multiresolution stream.

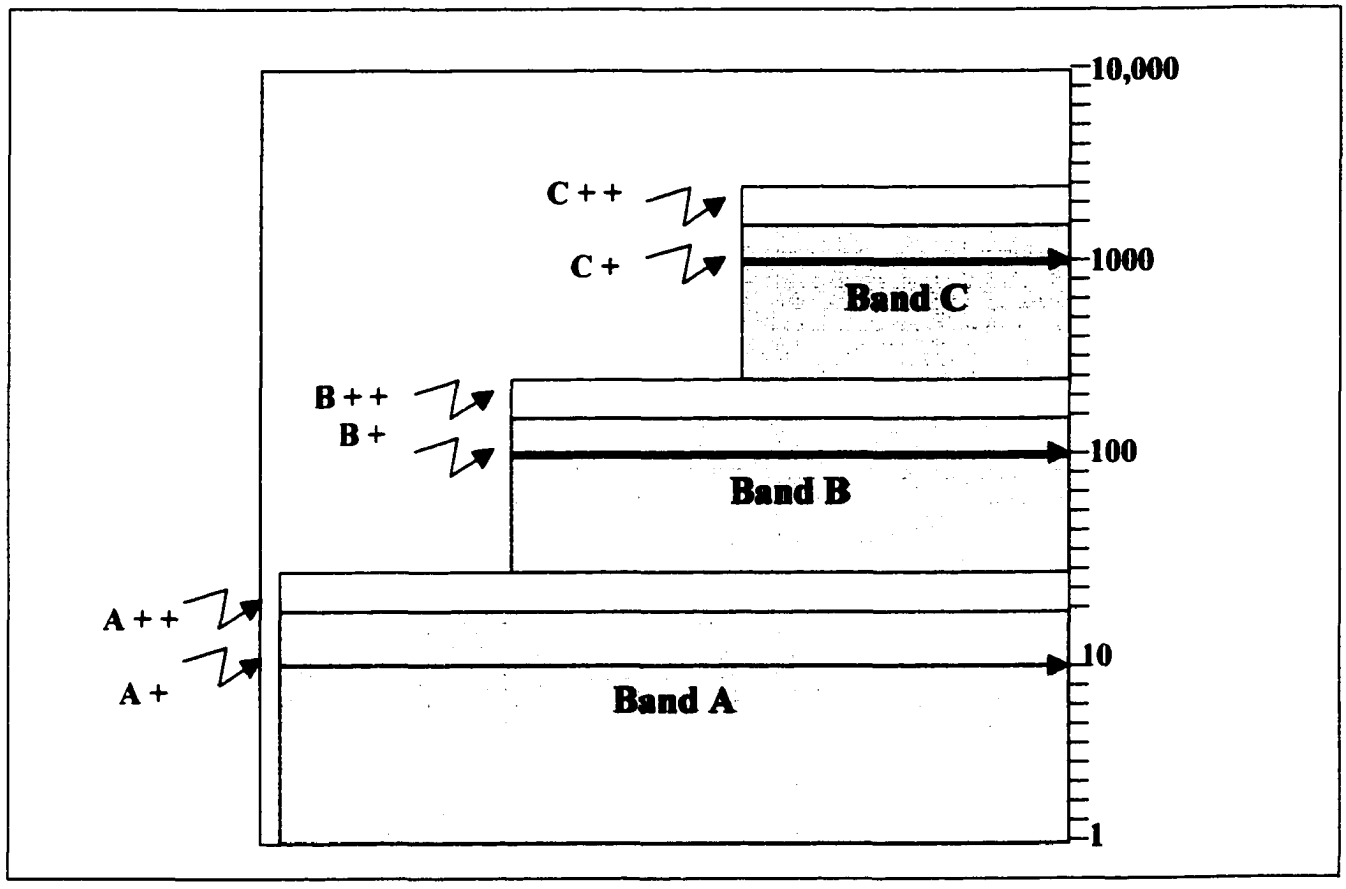


Fig 1.3 Layers of a multiresolution stream

Once this multiresolution stream is presented to the network, the network-based responsibility start to take place. In this framework, the network has a number of actions that can be executed at the transport layer routing and/or wireless link levels. These actions include: substream filtering, substream scheduling and error control, resource reservation, call admission and signaling.

Substream filtering is used at one or more mobility- and wireless-enabled switches or routers, and access points supporting a multicast connection associated with a mobile station as well

as in a mobile station. At every substream filter (SSF), packets corresponding to a substream are passed to (or blocked from) an outgoing link belonging to the multicast path according to the following criteria: The QoS requirement and priority of the substream defined by its SSI, the amount of bandwidth reserved for the connection and the available bandwidth on the outgoing link. Substream scheduling is executed at the mobile station to schedule the transmission of substreams according to their priority and QoS specifications. As a mobile station starts observing a short fading period on the wireless channel, it tries to prioritize the transmission of its substreams in order to achieve a minimum QoS. Here, depending on channel conditions, a substream (or layer) might be dropped for a period of time in order to accommodate higher-priority substreams. Thus, a scheduler provides a filtering function as well; however, its filtering function is a result of its scheduling function. It is important to note that the scheduler reacts to the fluctuations in the wireless channel due to error and fading conditions, and requires feedback from the wireless transmitter and receiver regarding wireless channel conditions to determine the state of the wireless channel and also to predict its near-term state.

In general, the paper provides an operational overview of the adaptive multimedia wireless/mobile network based on the concepts described in the paper. Example scenarios are shown under different conditions including: fast fading, slow fading or shadowing, handoff and resource re-allocation due to mobility and congestion. The paper presents a general framework for an adaptive QoS provisioning scheme for wireless/mobile networks showing all requirements and components (end-to-end) of such a system leaving the final details of the implementation up to the network design and implementation.

1.2.2.3 Terminal QoS for Adaptive Applications

In [12], a special focus on the adaptive applications is presented. The paper pays special attention to networked multimedia applications on slim hosts or terminals such as personal digital assistants (PDAs) and set tops. In addition to their limited processing capabilities, these terminals may use limited network bandwidth conditions—for example, wireless links. One way to implement multimedia applications on terminals is to make them adaptive. Such applications can then use bandwidth availability in the network as feedback to adapt their output packet rate. This manages the bandwidth scarcity by preventing the network congestion from worsening beyond a point. However, such an implementation imposes unique challenges. Simple task-level processing variations in audio/video applications can seriously overload a terminal. Adaptive applications impose even more variability. Consequently, the processing delay of applications can become unpredictable.

The adaptive application proposed in the paper is shown in Fig. 1.4. When the packet loss in the network grows beyond a threshold, the output frame rate of the video sender can be reduced so as to reduce the offered packet rate to the network, thereby possibly improving long-term queuing delays in the network. Thus, an adaptive video application can inherently exchange frame rate (perceptual quality) for network bandwidth. Audio applications can also be adaptive. Unlike video, audio can exchange compression for network bandwidth. The audio sender shown in Fig 1.4 implements several alternative compression algorithms within the encoder. If the packet loss in the network exceeds a threshold, the compression algorithm can be changed at run time in order to increase the amount of compression. For both audio and video applications, the loss feedback is conveyed using protocols such as real-time transport control protocol. Adaptive applications are "network friendly." They are especially

suited for use on terminals, since terminals are expected to be deployed in environments where network bandwidth is particularly scarce—for example, wireless links. However, adaptive networked applications offer a unique challenge from the perspective of system design. Each adaptation corresponds to a different computational demand. Thus, an adaptive application inherently offers a variable amount of computation. This makes it much more difficult to characterize the resource requirements on the terminal and to estimate the performance of an adaptive application. Moreover, since each adaptation of the application can affect the offered traffic rate to the network, the local performance of the applications on a terminal can affect the network performance. This, in turn, can affect future adaptations, giving rise to a complicated dependence relationship. This paper addresses these problems and makes two specific contributions. First, the paper proposes to quantify the local performance of applications on terminals in terms of their processing delay. Processing of audio and video applications at the sender or receiver end is usually driven by periodic deadlines, since the media source is sampled periodically. The processing delay estimate indicates the time taken to process each new sample across the terminal. The paper argument is that the application processing delay can affect network quality of service (QoS)—for example, packet loss. For this reason, it is (processing delay) classified as a QoS measure. The second contribution is an analytical framework that can be used to compute the terminal QoS.

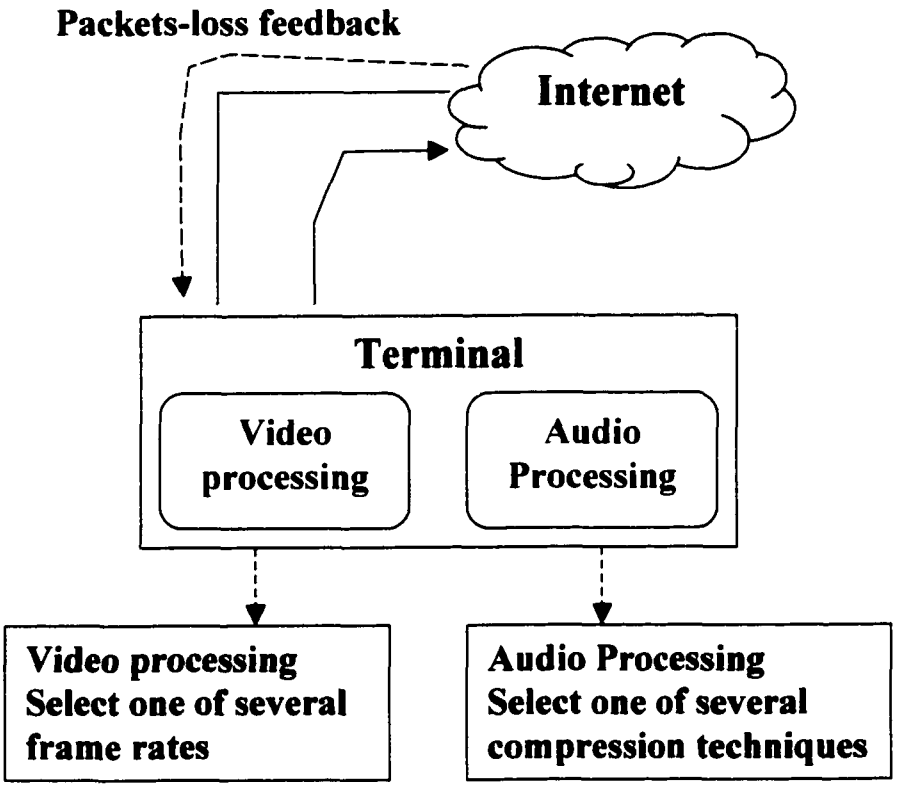


Fig. 1.4 Application that adapt to packet loss feedback from the network

In general, the paper presents some practical aspects with regards to the adaptive applications. But it does not address any network-based adaptation, or how the network will deal with the incoming multimedia streams. Furthermore, the approach and its analytical study is mainly suitable for mobile terminals with limited processing powers.

1.2.2.4 QoS-Aware Resource Management Scheme (QualMan)

In [13] and [14], a QoS-aware resource management scheme for distributed multimedia application is presented. The concept of adaptive QoS provisioning in this approach is quite different. In this approach, the QoS provisioning exists only at the end-points of the end-to-

end multimedia communication path. In other words, not all components along the end-to-end path (e.g., from video retrieval at a transmitting workstation to video display at a receiving PC) have QoS support or are even required to have any control or adaptation requirement over the multimedia traffic. The bottom line is that in this approach the QoS is application-controllable. In this paper, a new resource model and a time-variant QoS management, which are the major components of the QoS-compliant resource management. To achieve an end-to-end quality of service (QoS) along multimedia communication paths for distributed multimedia applications, there is a need for services and protocols in the end-points and networks which understand what quality of service is and how to map this quality into the required resource allocation. Furthermore, the underlying resource management must have services and protocols which know how to negotiate, admit, reserve, and enforce requested resource allocation according to requested QoS requirements. Since the approach depends mainly on the end-points, each end-point must be modeled autonomously enough to provide its own QoS control as well as being able to adapt to possible occurrences of non-deterministic system changes/overruns on general-purpose systems. The resource model incorporates, the resource scheduler, and a new component, the resource broker, which provides negotiation, admission and reservation capabilities for sharing resources such as CPU, network or memory corresponding to requested QoS. The resource brokers are intermediary resource managers; when combined with the resource schedulers, they provide a more predictable and finer granularity control of resources to the applications during the end-to-end multimedia communication than what is available in current general-purpose networked systems.

Furthermore, this paper presents the QoS-aware resource management model called QualMan, as a loadable middleware, its design, implementation, results, tradeoffs, and experiences.

In QualMan, admission control is dealt with in the following manner. Once the application specifies its communication QoS parameters at the time of connection setup, the broker performs checks to verify that the parameters can be guaranteed. The admission control mechanism, using an admission condition, decides if the requested QoS can be met or suggests a lower achievable value. The communication broker performs admission on bandwidth availability and end-to-end delays. Fig. 1.5 shows the role of the QoS broker during connection setup.

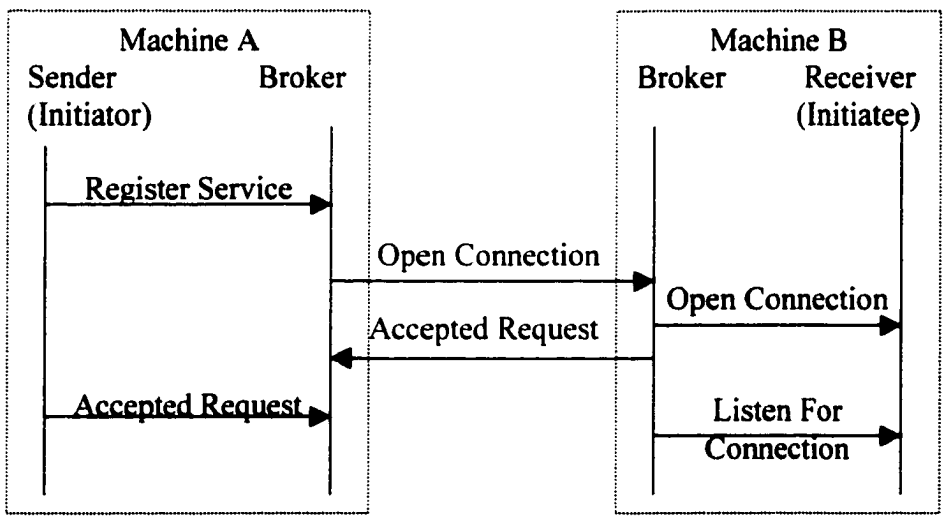


Fig. 1.5 QoS Broker during Connection Setup

The initiator sends the register request along with its QoS parameters. Second, the initiator broker performs admission control on the specified QoS. If the admission test is successful,

the initiator broker will send an open connection request to the initiatee side (broker), which also performs an admission test. If the admission test is also successful at the initiatee side, the initiatee will accept the connection setup, and the protocol will respond with an accepted allocation message. Upon receiving the accept message the initiator side completes the connection setup. Partial acceptance can also happen when the initiatee does not have available requested resources for end-to-end delay provision, and it sends back a message with partially fulfilled content (only bandwidth guarantees are given). The initiator of the QoS connection decides if this is sufficient. If this is the case, accepted allocation message is sent back to the initiatee, and a connection opens at the initiatee side with degraded quality. The third possibility is to send out a reject request message when bandwidth test is violated. Monitoring and adaptation are needed in order to allow for upgrading and degrading in the quality of connections. A monitoring thread examines the amount of available resources whenever a connection is closed. It checks if the freed resources can be used to satisfy any partially fulfilled connections. When such a connection is identified, the monitoring thread sends a message to its application over the register channel and informs it about the possible upgrade.

In general, the QualMan approach provides a quite interesting and different approach to adaptive QoS provisioning where only end-points are involved in the adaptation process. The advantages of QualMan is that it is flexible and scalable on a general-purpose workstations or PC. The disadvantage is the lack of very fine QoS granularity and details.

1.2.2.5 Peer-to-peer QoS Management

In [15] and [16], a QoS-based adaptation mechanism that is specially designed for multicasting communication is presented. It is required that future QoS mechanisms can ensure full quality media playout at high-performance workstations while at the same time providing appropriately filtered lower quality media for playout at low-end systems. Existing multicast support mechanisms are deficient for this purpose, in a heterogeneous environment, because they work on a lowest common denominator premise where the quality provided depends on the least capable link or node involved in the multicast session. In this paper a QoS model to provide receiver-dependent QoS based on filtering techniques is presented.

A model for the establishment and management of continuous media data flows between a single sender (source) and multiple receivers (clients) is presented. The QoS requirements of individual receivers are met through the utilization of filtering techniques. The filters operate on encoded data streams and can adapt a data flow to meet the special needs of single users. Experimental results concerning bit rate reductions included the use of the following techniques: dropper, color-to-monochrome transformation, transcoder and low pass filter. Real time requirements of the presented filters were also discussed. This included the performance of the parsers, mixers and droppers in terms of speed and suitability for real time applications. It was deduced that filtering operations are possible in a real-time environment, if at a slight cost in end-to-end delay. The saving on network throughput may in fact counteract the increase in network delay experienced.

In [17] and [18], more types of filtering techniques are presented in more details. These papers describe the implementation of a number of filtering mechanisms and highlight the communications architecture within which these mechanisms are built. These filters adapt the

multimedia streams to ensure that end-user, application, end-system, network capabilities and requirements are met. Examples of such filters include: (1) a codec filter that can be used to compress or decompress a bit-stream (2) a frame-dropping-filter that is used to reduce frame rates, (3) a frequency filter that performs operations on semi-uncompressed data including low-pass filtering and color reduction filter, (4) a mixing filter that is used to mix streams together or to multiplex audio and video. (5) a requantization filter that uses a larger quantization step as a method of rate reduction, and (6) a slicing filter that increases the number of slices in an MPEG stream per frame. These papers also discuss in details the options of filter location which is a key issue in the engineering of any filter model. An example of these filters is shown in Fig. 1.6.

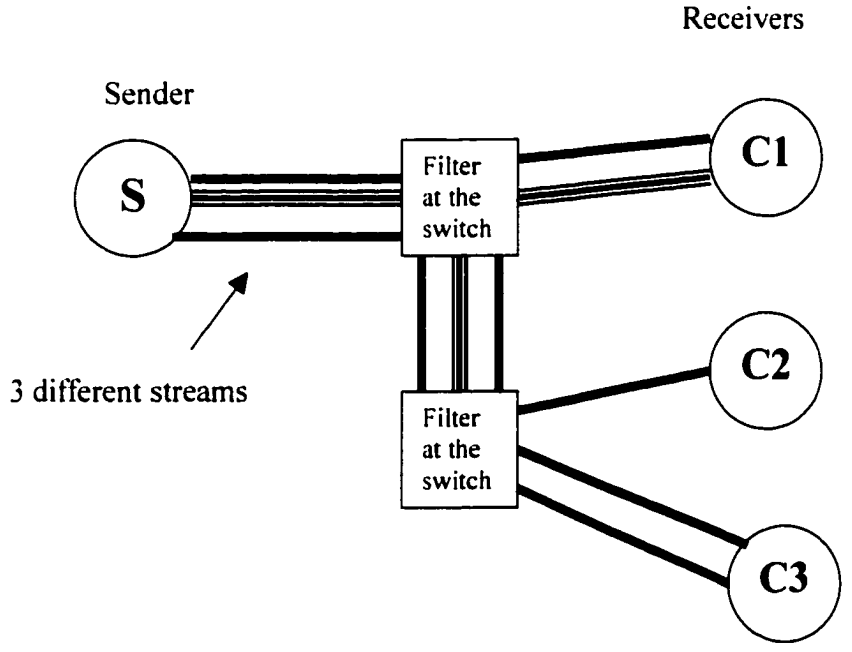


Fig. 1.6 Scaling Filter Example

In general, this approach is good for multicast networks where there is a single source of multimedia traffic and many receivers with diverse requirements and capabilities. Filters do not impose any requirements on the design of applications.

1.2.2.6 QoS Management for News-on-demand Applications

Ways in which an application can adapt to reduced network performance related throughput, loss, delay or jitter are discussed in [19]. The paper is based on the construction of a news-on-demand prototype. The paper is based on the construction of a news-on-demand prototype. This approach adopts the concept of having the network and the applications share the responsibility of providing an adaptive QoS. This approach introduces a number of network-based adaptation techniques. These techniques are: adaptive through alternative configurations and adaptive through alternative document structure. The idea of adaptation through alternative configurations is that in the case of network congestion the system tries to use another network. This may not be so easy, since most computers are nowadays only connected to a single network. However, in the case of an overloaded server computer, the switch over to another server may be a quite reasonable alternative. Alternative service providers are more common than alternative networks. The idea of adaptation through alternative document structures is to change the structure of the application, for instance, to replace some video stream by some corresponding text to be presented.

On the other hand, the user should be able to express the desired QoS depending on his needs and his financial capacity. The user's preferences are described in terms of (1) QoS setting for video, audio, still images and text, (2) the cost the user is willing to pay for a given quality, and (3) certain time constraints, such as the maximum acceptable delay for the presentation

of a news clip. These preferences must be described in terms of a set of user-perceived characteristics of the performance of the service. It should be a number of parameters.

Finally, the paper presents two main aspects: (1) the interface by which QoS characteristics can be negotiated with the user, and (2) the issues which are related to the adaptation of the application to the QoS characteristics. This paper uses the 'news-on-demand' as an example of the applications that allows users to browse through a remote database of news articles and retrieve selected articles for display on a workstation. A news article may consist of a single multimedia substream, such as test, a video clip, or a sound track, or may be a combination of suck substreams. In the prototype presented in the paper, the user may choose a document for presentation, and select the desired QoS including such parameters as video and audio quality, size of display, and cost. Several possibilities of QoS adaptation are also discussed in details.

In general, the paper follows the same approach of arguing that applications should be able to adapt to varying QoS characteristics that may be available from the underlying systems. However, no details of how these QoS characteristics are defined or how the user's view is translated to physical bit-rate has been provided. In addition, the approach presented is specially suitable for multicast communication or news-on-demand type communication. through alternative configurations.

1.2.2.7 Summary of Adaptive QoS Provisioning Schemes

Table 1.1 shows a summary of the adaptive scheme discussed in this section.

Table 1.1 Summary of Adaptive QoS Provisioning Schemes

Scheme	Network-based responsibilities	Application-based responsibilities	Main aspects of the scheme	Main shortage
DQM	✓	✓	- Dynamic Rate Shaping (application side) - Adaptation Network Service (Network side) - Extended application to wireless multimedia	- Lack of QoS mapping details - No details of real time application or admission control
End-to-end QoS Provisioning Framework	✓	✓	- Multiresolution Applications - Adaptive Network Components. - React to Mobility and Congestion - Comprehensive end-to-end framework	- Implementation details are left for network designers
Terminal QoS for Adaptive Applications	N/A	✓	- Focus on PDAs and set top boxes (terminals). - Adaptive Applications (video and audio) - Modeling the Terminal QoS	- Suitable for terminals with limited processing - No details about Network side
QualMan	X	✓	- Only end-points are involved - Resource Broker - Resource Scheduler - QoS is application controllable	- Lack of QoS Granularity - Lack of QoS mapping to real parameters
Peer-to-peer QoS	✓	X	- For Multicast communication - Excellent Filtering Techniques - No special application design	- Suitable for multicast communication only
QoS management (news on demand)	✓	✓	- Adaptation through network configuration and document structure - User interface to QoS requirements	- No details of QoS translation from user's view to requirements.

1.3 Proposed Approach

In ATM-based broadband networks, once a connection is admitted to the network, the set of resources used by that connection remains unchanged (static allocation approach). Furthermore, for connections that require guaranteed service, the call is admitted to the network if and only if a path through the network with enough resources can be established without violating the service guarantees of the existing ones. Once the network admits the connection, a contract between the network and the application is established whereby both try to adhere to throughout the connection lifetime. As long as the application does not violate its declared traffic parameters (e.g., peak bit-rate, burst length, etc.) the network should provide its committed QoS guarantees (cell loss rate, delay and jitter). Due to the statistical multiplexing effect and as the traffic increases, the resources allocated to a connection may temporarily fluctuate resulting in prolonged episodes of cell losses. Fig. 1.7 shows a block diagram of the different entities of the QoS management system. A new traffic source presents its parameters to the traffic conditioner which performs traffic shaping and policing. The conditioned traffic is then passed on to the traffic scheduler. The admission controller establishes a traffic contract with the new traffic source (new call). It also shares the contract information with the traffic conditioned and the scheduler.

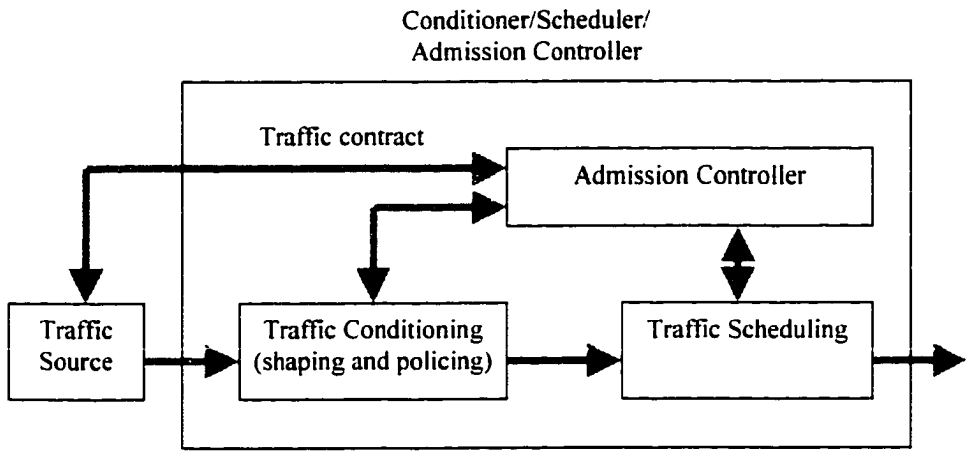


Fig. 1.7 Entities in a QoS Management System

As mentioned in the earlier section, in cellular based wireless networks, the QoS provisioning problem is more challenging due to wireless channel fading, bit error rate (BER) and mobility. During handoff, a mobile that was granted certain QoS guarantees could be deprived of such guarantees or even dropped altogether. Hence, the bandwidth allocated to a call, during setup phase, could be decreased significantly during the call lifetime. Therefore, it is desirable to adopt an adaptive allocation of resources approach for QoS provisioning. A call can respond to the fluctuations in the network resources by changing its bit rate, and also its requested QoS parameters. In this thesis, we adopted such adaptive approach.

1.4 Adaptive Allocation of Resources Algorithm

In the proposed adaptive networking environment, calls can be admitted to the system even if the available bandwidth is not sufficient to satisfy their highest QoS guarantees. To

accomplish this, the proposed algorithm will try to degrade the QoS levels of the existing calls in order to free some bandwidth. Each call is considered to have a pre-defined minimum QoS level (minQ) and a maximum QoS level (maxQ) that are defined in a user-defined profile (UDP). The (minQ) level corresponds to the minimum set of QoS requirements that the application is willing to tolerate. On the other hand, the (maxQ) level corresponds to the best QoS requirements a call can obtain whenever the bandwidth is available. In the case of a multimedia call, each of its substreams (i.e. video, audio and data) will have its own QoS level ranging from high to medium to low. For example, one user might be willing to tolerate a low video quality, but requires high audio quality and high speed data service. Such user will be granted an amount of bandwidth that corresponds to its declared minQ level. Whenever there is available bandwidth, the user might obtain an extra amount of bandwidth up to its declared maxQ level. It should be noted that the UDP approach also defines different "grades of service" at different costs that the users can subscribe to. For example, a network provider may choose to offer two grades of services with different cost structures. A subscriber to a premium service will pay high dollar for the call but will be allocated a UDP with the uppermost range of quality levels, whereas a subscriber to an economy service will have a UDP range of qualities that is significantly lower than the premium one.

In Fig.1.8 it is shown that when a call is being admitted to the network, it presents its traffic parameters and its declared UDP to the "traffic conditioner/scheduler/admission controller". The "traffic conditioner/scheduler/admission controller" will then calculate the required bandwidth according to the traffic parameters presented to it. The calculated bandwidth and the declared UDP are then saved in a database in order to be used by the proposed optimization algorithm as will be explained in chapter 2. Fig.1.9 shows an example of the

UDP where a call (i) is presenting both the maxQ and the minQ levels to the admission controller/scheduler. All QoS levels between the declared maxQ and minQ are allowed to be assigned to call i. In this example, the maxQ level corresponds to the highest video quality, the highest audio quality and the highest speed for data service. The minQ level corresponds to medium video quality, low audio quality and low speed for data service. Any QoS level between these declared levels are allowed to be allocated to this call i.

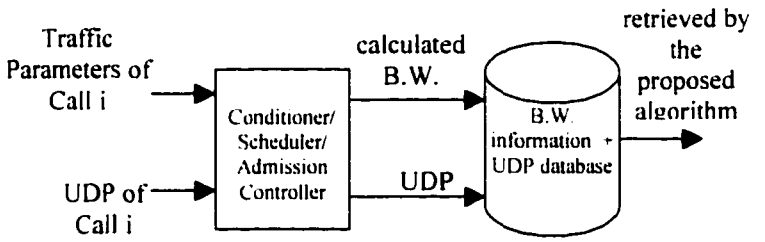


Fig.1.8 Database used by the algorithm

	Video	Audio	Data
maxQ	**High**	**High**	**High**
	High	High	Medium
	High	High	Low
	....	....	....
	....	....	....
	Medium	Medium	Low
minQ	**Medium**	**Low**	**Low**

Fig.1.9 Example of a User-Defined Profile (UDP)

1.5 QoS Levels for Multimedia Substreams

In the proposed algorithm, we use three different QoS levels for each of the video, audio and data substreams. Fig. 1.10 shows an example of the different QoS levels. These QoS levels can be updated and modified according to the multimedia applications used in the network. As shown in figure, the video component is the most dominant due to its high demand of bandwidth. The Figure also shows an example of the required bandwidth for each of the levels.

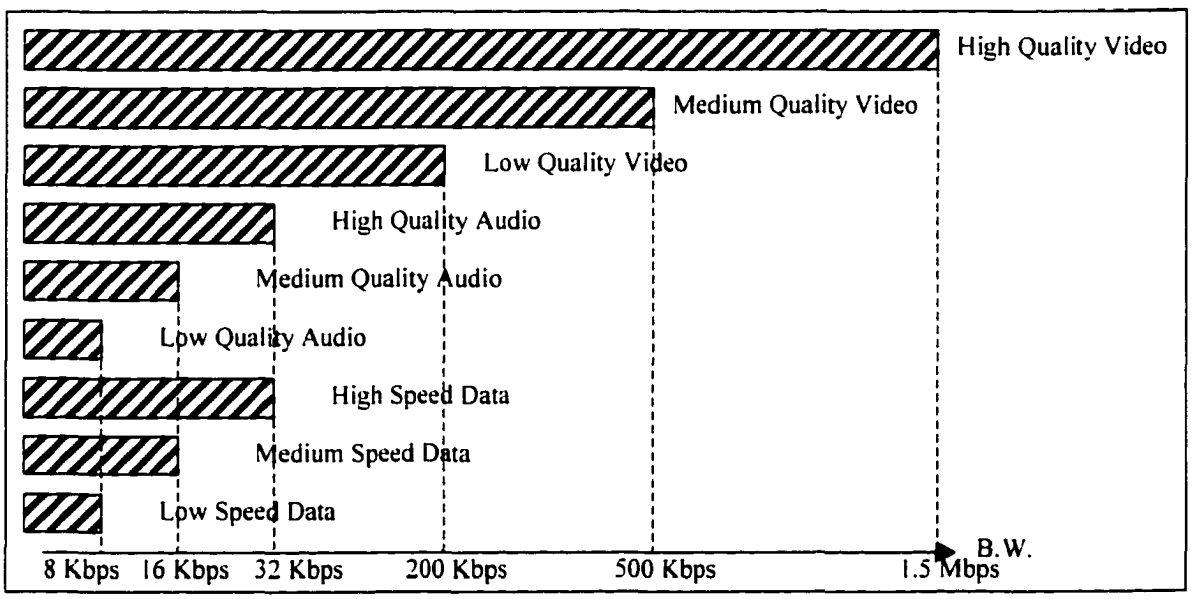
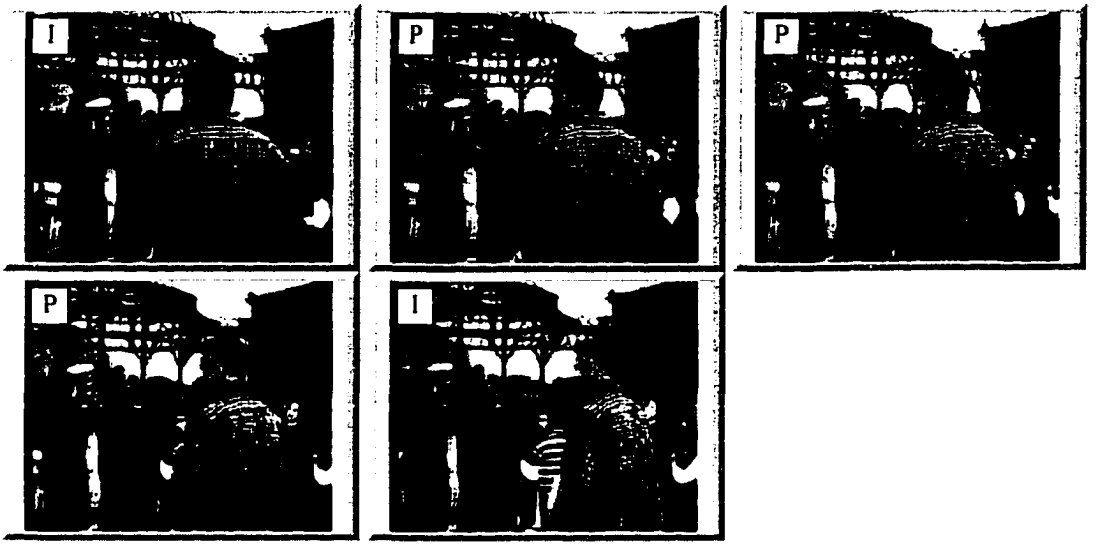


Fig.1.10 Example of QoS levels for video, audio and data

We use MPEG 1 video streams as an example of the video substream. For highest video quality, all frames (I,P and B) are transmitted; whereas for medium quality, only the I and P frames are transmitted. For low quality, only the I frames are transmitted. A similar scheme for MPEG was suggested in [10]. To show the practical effect of degrading the QoS levels of the MPEG streams from high to medium and from medium to low, we have used an MPEG

encoder/decoder and we captured 13 consecutive frames (*IBBPBBPBBPBB I*). Fig. 1.11 shows the captured frames for the different QoS levels (High, Medium and Low). As the Video QoS level is degraded from high to medium, B frames are dropped. The effect on the quality is that some discontinuities in the video sequence are noticed during play back. Whereas, for the low quality video, these discontinuities are clearer as only the I frames are transmitted. Obviously, such discontinuities are directly related to the degree of motion activities. Therefore, unless the degree of motion activity is high, the degradation in the quality is graceful.

(a) High QoS level for MPEG (*IBBPBBPBBPBB I*)



(b) Medium QoS level for MPEG (*I P P P I*)

(c) Low QoS level for MPEG (*I I*)

Fig.1.11 Live MPEG captured frames

1.6 Genetic Algorithms

Applications of artificial intelligence techniques in traffic management of ATM networks have proved to be a promising research area, see [20]-[25]. Special attention was paid for the Connection Admission Control (CAC) as an excellent candidate for artificial intelligence techniques [20],[25]. In the static approach, admission control implies that each new call

makes a request for a connection. The request includes the QoS required by the call. If the QoS can be met without deteriorating those of the existing calls, then the call is admitted, otherwise it is rejected (blocked). Thus an estimate of the QoS is required based on monitoring traffic patterns and buffer status. The number of cells waiting in the buffers for service is an important parameter for determining the cell loss probability, cell delay, and delay variations. A number of papers proposed a number of techniques to train neural networks to learn the average of those values and thus a more clear and precise QoS estimate can be calculated, see [20]. In summary, the application of neural networks to calculate a precise estimate of the QoS of a certain call by observing some of the traffic parameters related to this call, has proved to be both precise and robust. On the other hand, genetic algorithms have also been used to address diverse practical optimization problems related to traffic management, see [26]-[30]. In [29], a control scheme applied to the virtual path (VP) bandwidth allocation problem is presented. The concept of VP bandwidth allocation problem can be summarized as follows. A new connection is accepted only if the bandwidth of a chosen VP is sufficient or the request of increasing bandwidth is permitted. Each request will cause the bandwidth of the VP to increase by a specified step. In [29], a Neural-network-based genetic algorithm scheme is used to solve such a problem.

The proposed algorithm is based upon genetic algorithms (GAs) which belongs to the family of derivative-free optimization techniques. Other derivative-free methods include simulated annealing, random search and downhill simplex search [28]. Simulated annealing (SA) was derived from physical characterstics of metals and what happens when they are cooled at a controlled rate [28]. SA is known for its effectiveness in finding near optimal solutions for large-scale combinatorial optimization such as traveling salesperson problems. Random

search is the simplest optimization method, but is not suitable for large-scale problems. The downhill simplex search is suitable for multidimensional function optimization. Genetic algorithms search the solution space of a function through the use of simulated evolution, i.e., the survival of the fittest strategy. In general, the fittest individuals of any population tend to reproduce and survive to the next generation, thus improving successive generations. However, inferior individuals can, by chance, survive and also reproduce. Genetic algorithms have been shown to solve linear and nonlinear problems by exploring all regions of the state space and exponentially exploiting promising areas through mutation, crossover, and selection operations applied to individuals in the population [27].

Genetic algorithms rely on repeated evaluation of the objective function. This makes GAs more attractive as an optimization tool since it does not need functional derivative information. Furthermore, GAs are parallel-search procedures that can be implemented on parallel processing machines. GAs have been used quite successfully to address diverse practical optimization problems (see [27],[30]).

The genetic algorithm starts from a set of individuals (assumed solution set for the function to be optimized) and proceeds from generation to generation through genetic operations. Replacement of one or more individuals at a time is called reproduction. Reproduction includes two main operations, crossover and mutation. Genetic algorithms require only a suitable evaluation function which acts as the environment in order to evaluate the suitability of the derived solutions (chromosomes).

In its most fundamental form, called simple genetic algorithm (SGA) [27], the genetic algorithm is can be summarized in the following pseudocodes:

```
GA()
{
    generate randomly an initial population;
    evaluate fitness of each individual;
    while convergence not achieved and max. number of generations not exceeded
    {
        select individuals probabilistically according to their fitness;
        perform genetic operations on the selected individuals;
        evaluate fitness of the newly obtained individuals;
    }
}
```

The use of a genetic algorithm requires the determination of six fundamental issues: chromosome representation, selection function, the genetic operators making up the reproduction function, the creation of the initial population, termination criteria, and the evaluation function. These issues are now described.

1) Solution (chromosome) Representation: In the SGA, a possible solution for the optimization problem is called an individual. A set of individuals represents a group of candidate solutions called a population. For any GA, a chromosome representation is needed to describe each individual in the population of interest. The representation scheme determines how the problem is structured in the GA and also determines the genetic operators that are used. Each individual or chromosome is made up of a sequence of genes from a certain alphabet. An alphabet could consist of binary digits (0 and 1), floating point

numbers, integers, symbols (i.e., A, B, C, D), matrices, etc. Each individual is usually represented by a binary vector. However, it has been shown that more natural representations are more efficient and produce better solutions [31]. One useful representation of an individual involves variables from an alphabet of floating point numbers with values within variables upper and lower bounds. In [31], extensive experimentation comparing real-valued and binary GAs was done. The real-valued GAs were proved to be an order of magnitude more efficient in terms of CPU time. It was also shown that a real-valued representation moves problem closer to the problem representation which offers higher precision with more consistent results across replications. Therefore, in our implementation we used the floating point representation.

2) Selection Function: The selection of individuals to produce successive generations plays an important role in a genetic algorithm. A probabilistic selection is performed based upon the individual's fitness such that the better individuals have an increased chance of being selected. An individual in the population can be selected more than once with all individuals in the population having a chance of being selected to reproduce into the next generation. Each individual here represents a candidate solution to the problem. The fitness of each individual is determined using the evaluation function discussed later.

A common selection approach assigns a probability of selection, P_j , to each individual, j based on its fitness value [32]. A series of N random numbers is generated (N is the population length) and compared against the cumulative probability, C_i , of the population.

$$C_i = \sum_{j=1}^{i} P_j \qquad (1.1)$$

The appropriate individual, i, is selected and copied into the new population if

$$C_{i-1} < U(0,1) \leq C_{i-1} \qquad (1.2)$$

Various methods exist to assign probabilities to individuals [32]: Roulette wheel, linear ranking and geometric ranking. In our implementation, we use the geometric ranking selection function. This method only requires the evaluation function to map the solutions to a partially ordered set, thus allowing for minimization and negativity. Geometric ranking method assigns P_i based on the rank of solution i when all solutions are sorted. P_i for each individual is defined by:

P[selecting the i^{th} individual] = $q' (1 - q)^{r-1}$ (1.3)

Where

q = the probability of selecting the best individual,

r = the rank of the individual, where 1 is the best,

P = the population size,

$$q' = \frac{q}{1-(1-q)^{P}} \quad (1.4)$$

3) Genetic Operators: Genetic Operators provide the basic search mechanism of the GA. The operators are used to create new solutions based on existing solutions in the population. There are two basic types of operators: crossover and mutation. Crossover takes two individuals and produces two new individuals while mutation alters one individual to produce a single new solution. The application of these two basic types of operators and their derivatives depends on the chromosome representation used [31],[32].

Let A and B be individuals (parents) from the population. For binary A and B, the following operators are defined: simple crossover and binary mutation. Simple crossover generates a random number r from a uniform distribution from 1 to m and creates two new individuals (a' and b') according to the following two equations.

$$\grave{a}_i = \begin{cases} a_i, \text{ if } i < r \\ b_i, \text{ otherwise} \end{cases} \quad (1.5)$$

$$b'_i = \begin{cases} b_i, \text{ if } i < r \\ a_i, \text{ otherwise} \end{cases} \quad (1.6)$$

Where i is the binary bit location, m is the length of the binary word representing each individual. $\grave{a}_i$ is the ith bit in individual A after mutation, and a_i is the ith bit before mutation. Other types of crossover include uniform crossover and linear interpolation n-point crossover.

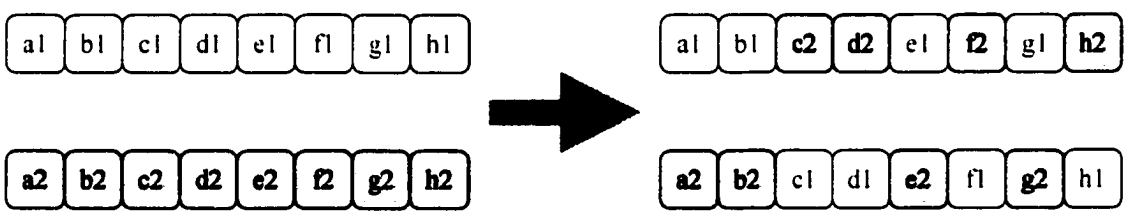


Fig. 1.12 Example of crossover operation

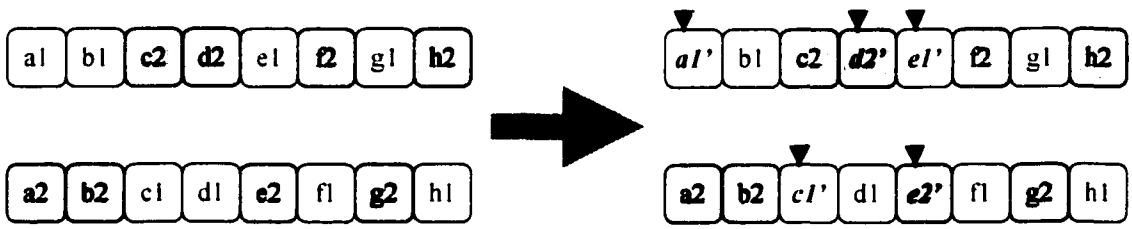
Fig. 1.12 shows an example of the crossover operation where two individuals reproduce using the crossover operation resulting in two new individuals. The 8 bits representing chromosome 1 are represented here as "a1 b1 c1 d1 e1 f1 g1 h1". The 8 bits representing chromosome 2 are represented here as "a2 b2 c2 d2 e2 f2 g2 h2". Chromosomes 1 and 2 represent the two parents chosen for the crossover operation.

After the application of the crossover operator, each of the chromosomes is subject to possible mutation [31],[32]. Binary mutation flips each bit in every individual in the population with probability p_m, the mutation rate, according to the following equation.

$$\grave{a}_i = \begin{cases} 1\text{-}a_i, \text{ if } U(0,1) < p_m \\ a_i, \text{ otherwise} \end{cases} \qquad (1.7)$$

Where i is the binary bit location, m is the length of the binary word representing each individual, $\grave{a}_i$ is the ith bit in individual A after mutation, a_i is the ith bit before mutation, and U(0,1) is the probability of bit i being mutated (according to a certain probability distribution). More sophisticated options apply mutation to a subset of the individuals in each population according to another probability p_c [31],[32].

An example of a mutation operation, applied to the resulting chromosomes after crossover, is shown in Fig. 1.13.



▼ Complemented bit.

Fig. 1.13 Example of mutation operation

Operators for real-valued representations, i.e., floating point, were developed in [31],[32]. For real X and Y, the following operators are defined: uniform mutation, non-uniform

mutation, multi-non-uniform mutation, boundary mutation, simple crossover, arithmetic crossover, and heuristic crossover.

Non-uniform mutation and arithmetic crossover operations were used in our implementation. Let x_i and y_i be the lower and upper bound, respectively, for each variable i. Non-uniform mutation randomly selects one variable, j, and sets it equal to a non-uniform random number:

$$\grave{a}_i = \begin{cases} a_i + (y_i - a_i)f(G), & \text{if } r_1 < 0.5, \\ a_i + (a_i - x_i)f(G), & \text{if } r_1 \geq 0.5, \\ a_i, & \text{otherwise} \end{cases} \tag{1.8}$$

Where,

$$f(G) = (r_2(1 - \frac{G}{G_{max}}))^s, \tag{1.9}$$

r_1, r_2 = a uniform random number between (0,1),

G = the current generation,

G_{max} = the maximum number of generations,

s = a shape parameter.

4) Termination: The GA moves from generation to generation selecting and reproducing parents until a termination criterion is met. The most frequently used stopping criterion is a specified maximum number of generations. Another termination strategy involves population convergence criteria. In general, GAs will force much of the entire population to converge to a single solution. When the sum of the deviations among individuals becomes smaller than some specified threshold, the algorithm can be terminated. The algorithm can also be terminated due to a lack of improvement in the best solution over a specified number of

generations. Alternatively, a target value for the evaluation measure can be established based on some arbitrarily "acceptable" threshold. Several strategies can be used in conjunction with each other.

5) Initialization: The GA must be provided an initial population as indicated in step 1 of the pseudocode. The most common method is to randomly generate solutions for the entire population. However, since GAs can iteratively improve existing solutions (i.e., solutions from other heuristics and/or current practices), the beginning population can be seeded with potentially good solutions, with the remainder of the population being randomly generated solutions. The size of the population should be a fraction of the total number of individuals (possible solutions). There is no optimum population size. In our implementation (shown in chapter 2), for the initial population, random seeds are used in module I (in addition to the values fed back from the previous allocation), while excellent seeds are selected for module II.

6) Evaluation: Each individual is evaluated to obtain its fitness function F which will be defined for each of the two modules. The fitness function evaluates the closeness of a certain individual (solution) to the optimum solution. The fitness function should be carefully designed to describe the optimization problem.

The rest of the thesis is organized as follows. In chapter 2 we present the proposed adaptive allocation of resources algorithm. This includes the details of the genetic algorithm. The proposed algorithm is divided into two main modules. Both modules are also discussed in details in two sections. In chapter 3, simulation results are presented and discussed in details. Chapter 4 presents a proposed QoS based channel borrowing algorithm. This algorithm is

aimed at enhancing the overall performance of the proposed adaptive algorithm presented in chapter 2. Chapter 4 also includes detailed simulation results of a cellular based wireless system. Finally, chapter 5 concludes the work in this thesis and presents some possible future extension of this work.

Chapter 2: Adaptive Allocation of Resources Algorithm

In this chapter, we present the details of the proposed adaptive allocation of resources algorithm. As introduced in chapter 1, we assume that our system architecture allows four different QoS levels for each of the three multimedia substreams. Table 2.1 shows a possible scenario to define multiple QoS levels. The table defines the maximum QoS level to have an index of 1 (high resolution video, high quality audio and high speed data), whereas index 64 implies that the call has been dropped (no video, audio or data components). A user subscribing to "premium service" for calls having all three substreams (video, audio and data) will have a UDP with a QoS index that belongs to the following set of indices {1,2,5,6,17,18,21,22}. The "premium service" can also be defined for calls having partial substreams (e.g., audio plus data, video plus audio, video plus data, etc.). For example levels 49,50,53 and 54 are the "premium service" for applications that do not need video (i.e., audio plus data calls). Similarly, levels 61 to 62 are the premium service for data applications. Another type of service named "economy service" can be defined from Table 2.1. For calls having all three substreams, levels 33 to 35 and 37 to 39 will represent this service, which require minimal (Low) video quality. Similarly a "low priority service" can be defined using the levels ranging from 41 to 43. Fig.2.1 shows the bandwidth required to support 24 multimedia calls at different quality levels ranging from 1 to 64. The figure shows that there is a drop in the bandwidth required when the QoS index drops from 16 to 17 and also from 32 to 33 and from 48 to 49. This is because the video substream demands more bandwidth than the other two (audio and data). Hence, the drop in the bandwidth required occurs when the video quality drops from high to medium (QoS index 16 down to 17), and from medium

to low (QoS index 32 to 33), and from low to null (QoS index 48 to 49). Also the figure reveals that the 64 indices could be reduced into a smaller set. For example, indices ranging from 1 to 16 could be combined into only 4 or 5 distinct indices, whereas those from 17 to 32 could be combined into a single index. Similarly, those from 33 to 48, and from 49 to 64, each could be combined into a single index. This would significantly reduce the size of the information needed to be stored in the database.

For N calls in the system, we define the set $\mathbf{[Q]}=[Q_1,Q_2,Q_3,Q_4,\ldots Q_{N-2},Q_{N-1},Q_N]$, where Q_i is the QoS index for call i and $Q_i \in \{1,2,\ldots 64\}$. The set $\mathbf{[B]}=[B_1,B_2,B_3,B_4,\ldots B_{N-2},B_{N-1},B_N]$ represents the set of bandwidth requirements. The problem now is to find the best QoS levels for all existing calls amid a large search space. For N=10 calls, the search space will be $64^{10}=1.153 \times 10^{18}$ different combinations. The optimization problem can then, be defined as finding the set of QoS levels **[Q]** that corresponds to the set of bandwidths **[B]** such that

$$\sum_{i=1}^{N} B_i \leq Cav \text{ and } (Cav - \sum_{i=1}^{N} B_i) \text{ is minimum} \qquad (2.1)$$

where Cav is the available capacity in the current cell

In other words, we are searching for the best possible QoS level for each of the existing calls. The sum of bandwidths corresponding to those QoS levels is the closest to the maximum capacity (in order to maximize the utilization).

Table 2.1 Translation table for the QoS indices

QoS Index	Video	Audio	Data
1	H	H	H
2	H	H	M
3	H	H	L
4	H	H	N
5	H	M	H
6	H	M	M
7	H	M	L
8	H	M	N
9	H	L	H
10	H	L	M
11	H	L	L
12	H	L	N
13	H	N	H
14	H	N	M
15	H	N	L
16	H	N	N
17	M	H	H
18	M	H	M
19	M	H	L
20	M	H	N
21	M	M	H
22	M	M	M
23	M	M	L
24	M	M	N

QoS Index	Video	Audio	Data
25	M	L	H
26	M	L	M
27	M	L	L
28	M	L	N
29	M	N	H
30	M	N	M
31	M	N	L
32	M	N	N
33	L	H	H
34	L	H	M
35	L	H	L
36	L	H	N
37	L	M	H
38	L	M	M
39	L	M	L
40	L	M	N
41	L	L	H
42	L	L	M
43	L	L	L
44	L	L	N
45	L	N	H
46	L	N	M
47	L	N	L
48	L	N	N

QoS Index	Video	Audio	Data
49	N	H	H
50	N	H	M
51	N	H	L
52	N	H	N
53	N	M	H
54	N	M	M
55	N	M	L
56	N	M	N
57	N	L	H
58	N	L	M
59	N	L	L
60	N	L	N
61	N	N	H
62	N	N	M
63	N	N	L
64	N	N	N

H	High
M	Medium
L	Low
N	No Component

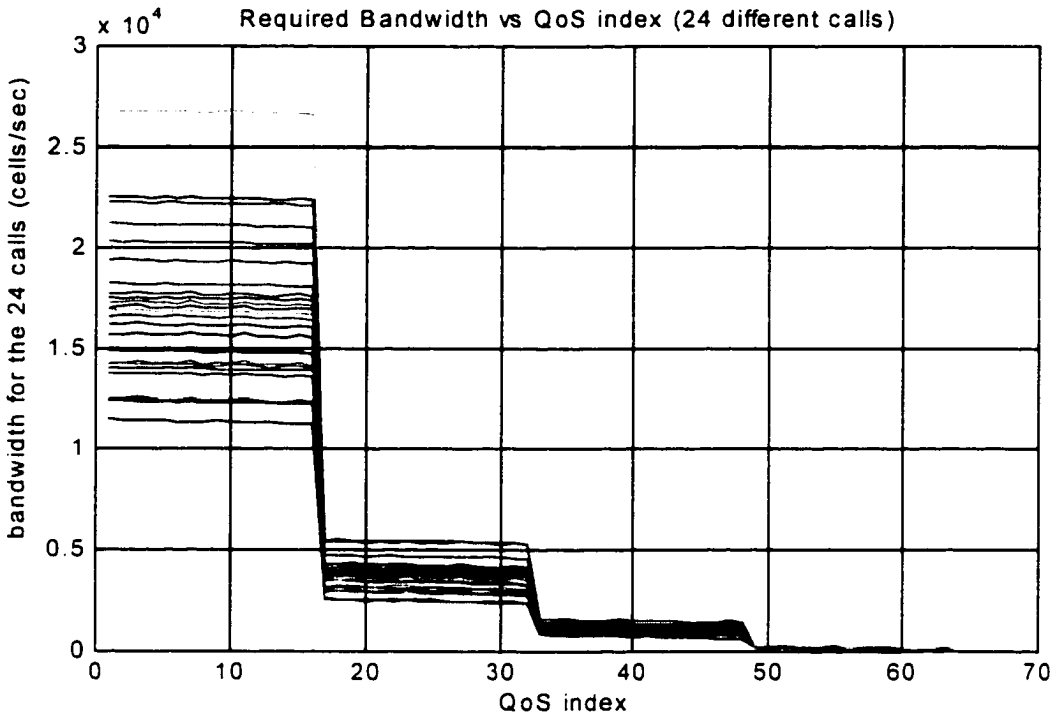


Fig.2.1 Required Bandwidth versus QoS indices

Fig.2.2 shows the block diagram of the proposed allocation of resources algorithm. As shown, the algorithm consists of two main modules: genetic algorithm modules I and II. The reason to use two modules is to divide the problem into smaller easier to handle sub-tasks since the search space is too large for a single GA. Module I assigns fair bandwidth allocations to the existing calls; whereas module II maximizes the capacity utilization by assigning any available bandwidth (left over from module I allocations) to the existing calls. Eventually, the system will not force a call to drop unless there is not enough bandwidth to satisfy its minQ level.

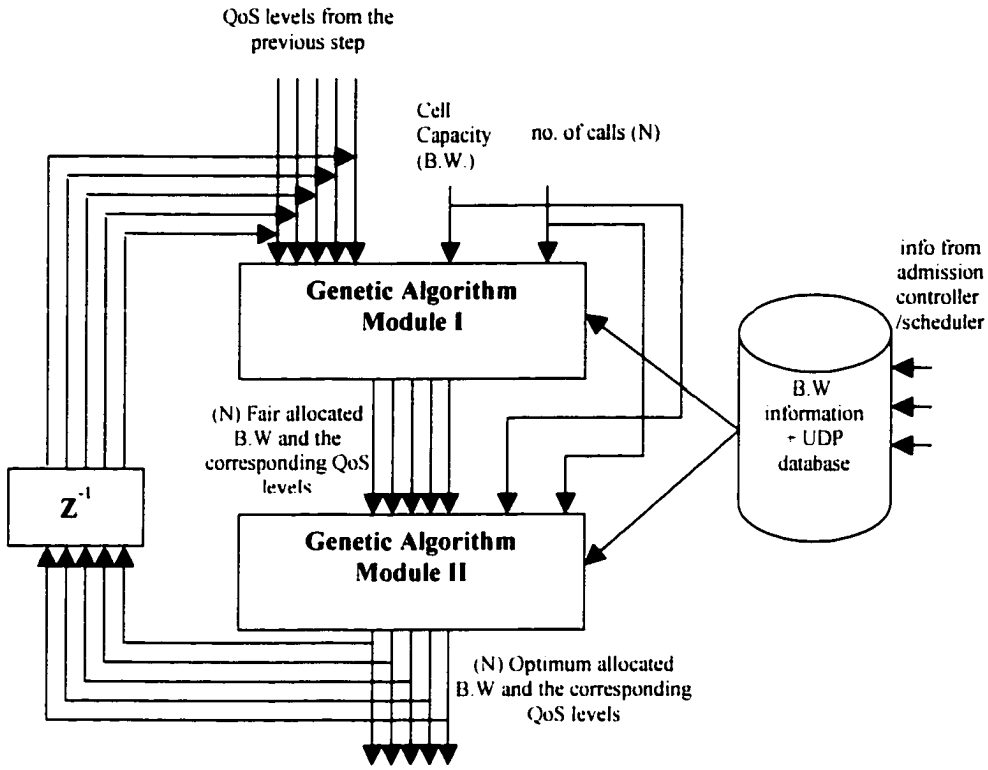


Fig.2.2 Block Diagram of the Proposed Allocation of Resources Algorithm

Module I is triggered whenever a call arrival or departure takes place. The inputs to module I include: (1) the capacity of the system (physical cell capacity), (2) the number of calls (N) after the arrival/departure and (3) the bandwidth information for these calls and their preset UDPs (which are fetched from the database). Furthermore, QoS levels from the previous allocation are also fed back to module I.

Module I searches for the QoS levels that correspond to the bandwidth allocation closest to the "fair" bandwidth. The evaluation of the optimization function is based upon the following rules:

$$B_{fair} = \frac{Cav}{N} \quad (2.2)$$

where

B_{fair} Fair Bandwidth Allocation for call i

Cav Total Physical Cell Capacity

In Eq.2, the calculation of the fair bandwidth is applicable to calls of similar classes (i.e., having the same type of multimedia substreams).

For each of the existing calls:

$$B_{modI(i)} \in B_i(Q_i) \quad (2.3)$$

$$B_{modI(i)} = \min(\text{absolute}(B_{fair} - B_i(Q_i))) \quad (2.4)$$

$$B_{modI(i)} \leq B_{fair} \quad (2.5)$$

Where

i	The call number.
$B_{modI(i)}$	The bandwidth requirement corresponding to the output QoS level from module I.
$B_i(Q_i)$	The bandwidth requirement corresponding to the QoS level Q
B_{fair}	The fair bandwidth

Once the fair allocations are determined by module I, module II is triggered. Module II redistributes any available capacity left over from module I among the existing calls. The evaluation of the optimization function for module II is based upon the following rules:

$QmodII \in [Q]$ (2.6)

For each call i,

$BmodII(i) = Bi(QmodII(i))$ (2.7)

$BmodII(i) \geq BmodI(i)$ (2.8)

Where

i	Call number,
[Q]	Set of QoS levels,
QmodII	A vector of N QoS levels (output of Module II),
BmodII	A vector of N Bandwidth requirements corresponding to Q modII,
BmodI	Bandwidth requirement resulting from module I.

The last rule is applied to ensure that the allocated bandwidth is greater than or equal to the "fair" one. Thus each call will at least be granted its fair share of the bandwidth allocation. Furthermore, this adds the advantage of decreasing the search space since the search will be focused only among QoS combinations having indices greater than or equal to those found by module I. Once module II finds the optimum solution, the QoS indices are assigned to the existing calls. This output is fed back to module I to be a subset of the initial population (candidate solutions) for the next allocation. Hence, module I will have an initial population that is in the vicinity of the optimum solution. This will significantly reduce the processing time in the next allocation cycle.

2.1 Genetic Algorithm Module I

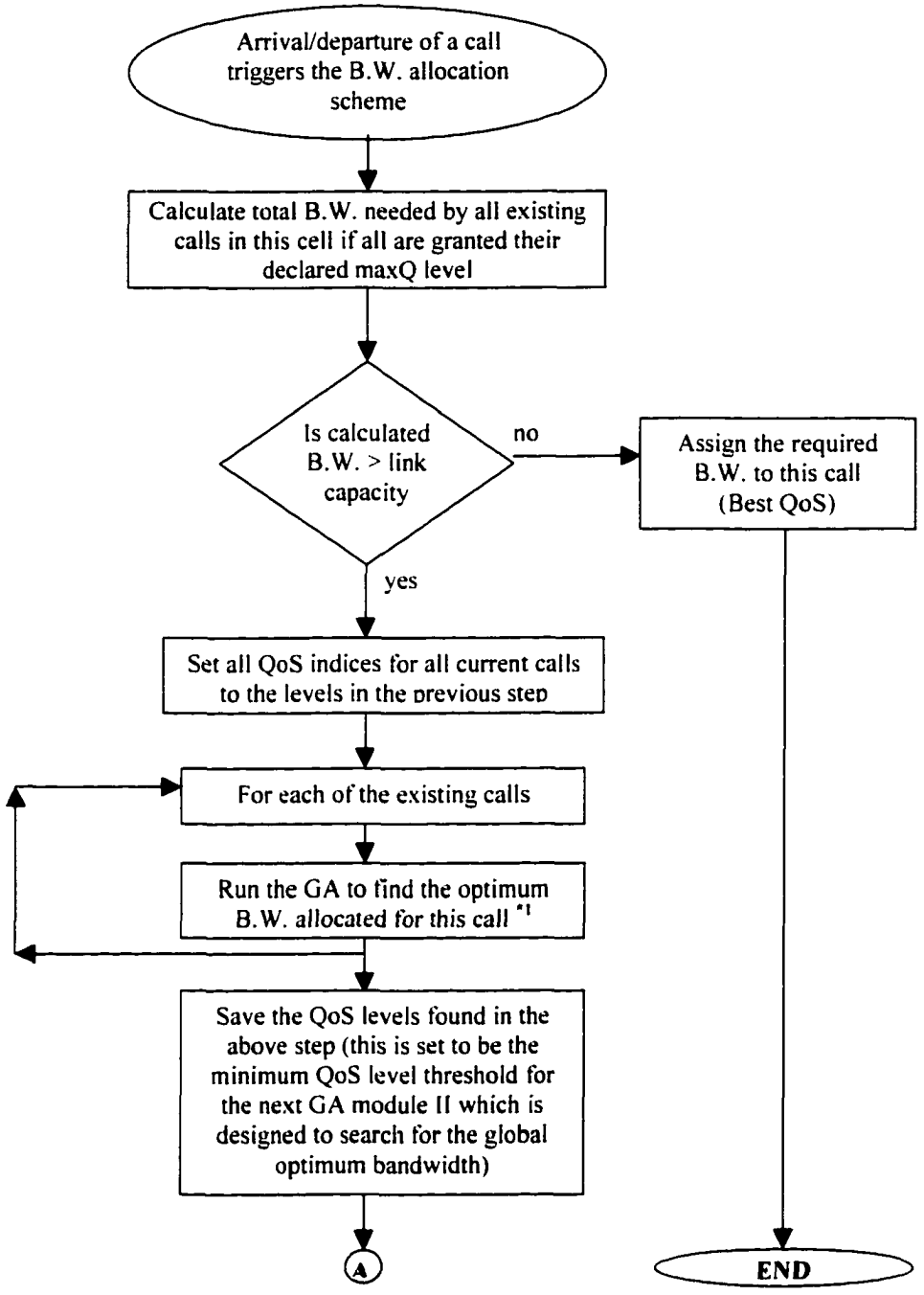
As previously discussed, module I is dedicated to allocating fair bandwidth shares. The fitness function for the GA used in this module is shown below.

```
Fitness_function_GA1(call_number,QoS_index)
if bandwidth(call_number,QoS_index)>fair_bandwidth
        return(fair_bandwidth - bandwidth(call_number,QoS_index));
else
        return(bandwidth(call_number,QoS_index) - fair_bandwidth);
end
```

The fitness function shown above assumes that *bandwidth(call_number,QoS_index)* is the bandwidth need by this *call_number* if granted a QoS level equal to *QoS_index*. Fig. 2.3 shows a detailed flowchart for module I.

2.2 Genetic Algorithm Module II

This module is designed to take advantage of any available bandwidth left over from module I. It tries to maximize the link capacity utilization and thus maximizing the QoS levels for the calls. In order to obtain an efficient solution we proposed three features. (1) the generation of the initial population, (2) the fitness function for the GA used in module II and (3) a decomposition scheme that is invoked if the number of calls is too large.



*1 – Optimality criteria: Search for the QoS level that corresponds to a bandwidth that is the closest to the "fair" bandwidth (Cell Capacity/total number of calls), but must also be less than or equal to this "fair" bandwidth

Fig.2.3 Genetic Algorithm Module I

The output of module I is just one possible solution (that is intended to be fair). Other excellent solutions (initial seeds) for the initial population can be found in the vicinity of this solution. Each member of the GA population is a vector of QoS indices of size N for the N calls present in the system at the time module II is triggered. One possible scheme to find excellent seeds for the initial population is shown below.

Let us denote the QoS indices, for the existing calls, resulting from module I as "seed_init".

```
Initialize_population(pop_length)
{
init_pop(1)=fitness_function_GA2(seed_init);
for I=2:(max_QoS_index/2)
        init_pop(I)=fitness_function_GA2(seed_init .-(I+1));
for I=(max_QoS_index/2):pop_length
        init_pop(I)=fitness_function_GA2(random(N) .< seed_init);
}
```

The (.-) denotes conditional vector subtraction. If the result of the vector subtraction is less than 1, the result is set to 1. Also *(random(N) .< seed_init)* denotes the generation of N (number of calls) random QoS indices (seeds) that are less than those of module I (QoS index of 1 is the highest QoS level and 64 (*max_QoS_index*) is the lowest).

The fitness function for the GA used in this module is shown below.

```
fitness_function_GA2(population_member)
{
total_bandwidth=0;
for I=1:N
        total_bandwidth=total_bandwidth+bandwidth(call_number,population_member(I));
if total_bandwidth>Link_capacity
        return((Link_capacity-total_bandwidth)/Link_capacity);
else
        return(total_bandwidth/Link_capacity);
end
}
```

The variable *population_member* is a vector of N QoS indices.

Another contribution to our algorithm is a decompostion scheme that is used mainly to decompose the search space into smaller subsets. Doing so fascilitates the job of the genetic algorithm in finding the best solution. The decomposition scheme is only triggered if the number of calls exceed a certain limit beyond which the genetic algorithm will either need more time to find the best solution or stop after a number of iterations without guaranteeing the solution to be the best one. In our simulation, it was found that as long as the number of calls were below 10 calls, the genetic algorithm was able to search and find the best solution. When the number of calls exceeded 10, the decomposition scheme is triggered. The motivation to use a decomposition scheme is to limit the search space, since if it is too large

the output will not be the best possible solution. As shown in Fig. 2.4, the decomposition scheme divides the population into a number of partitions Np. The scheme tries to find the best solution within each partition. In a second stage, it will then find the best solution among those that were selected from each partition. Finally that solution will be the one that maximizes the utilization and simultaneously corresponds to the maximum possible QoS level for each call. Each partition is of size (partition_width X population_length). From simulation, we found that a partition_width of 3 calls and a population_length of 1000 individuals yielded good results, causing the GA to converge quickly. In general, for a large number of calls where N>20, the following equation is used to calculate the number of partition.

$$Np=N-partition\_width+1 \qquad (2.9)$$

However in our case, since N was less than or at most 14 we obtained good results using Np=2.

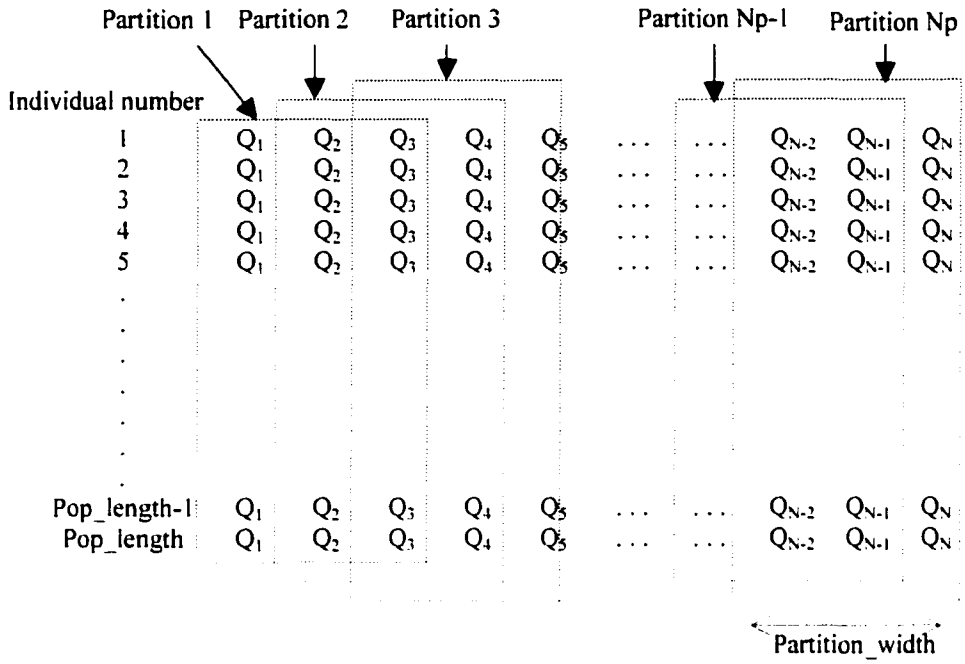
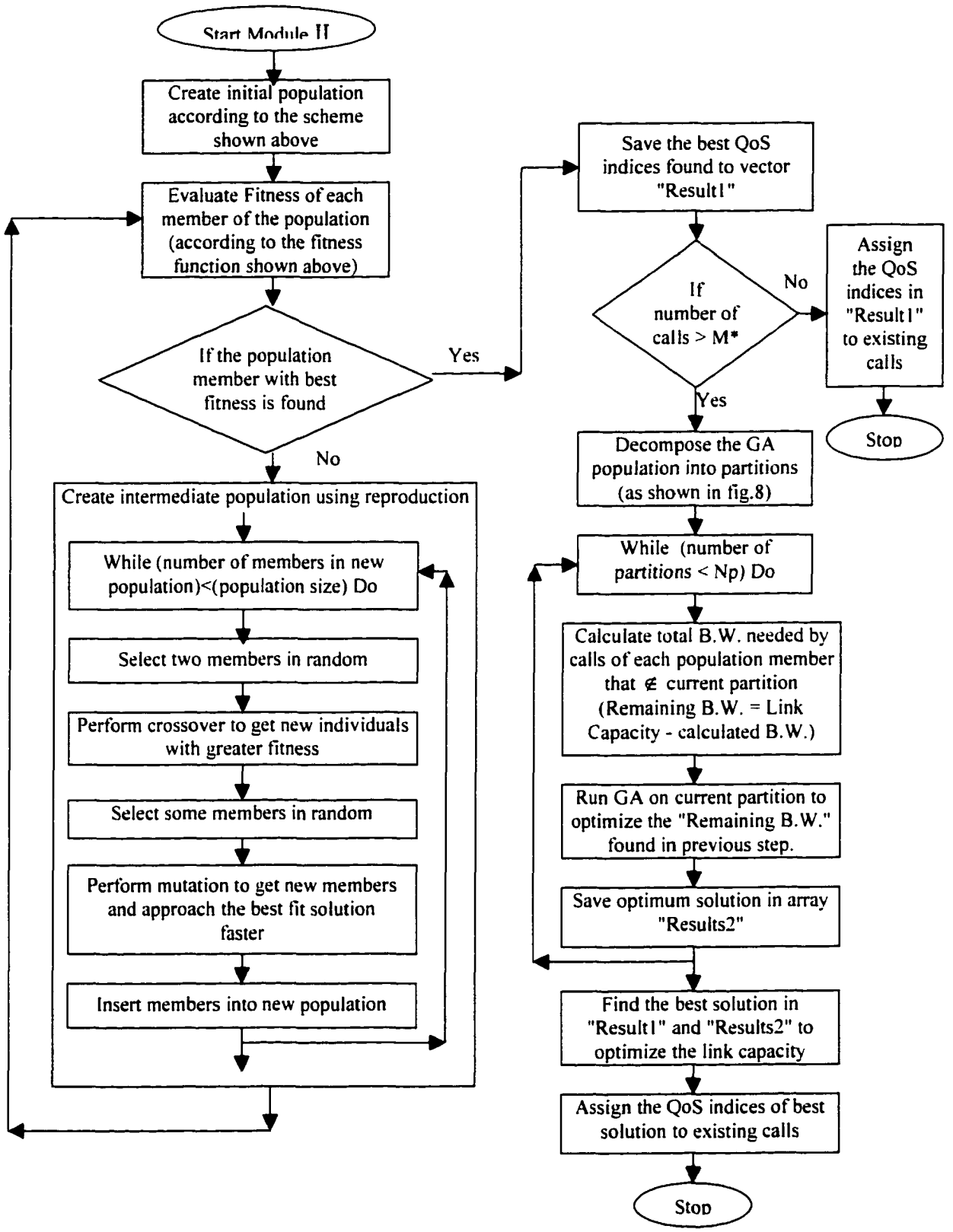


Fig.2.4 Decomposition Algorithm

This has resulted in decreasing the search space in each of these partitions to 64^3 instead of 64^N. Simulation results show that the decompostion scheme was very effective in decreasing the time needed by the GA to find the solution and also in guaranteeing its convergence.



* The number M depends on the computational speed of the GA and the population length.

Fig.2.5 Genetic Algorithm Module II

Fig. 2.5 shows a detailed flowchart of the genetic algorithm module II including the decomposition scheme and its interaction with the rest of the components of module II.

The next chapter presents a detailed simulation study of the proposed algorithm.

Chapter 3: Simulation Results

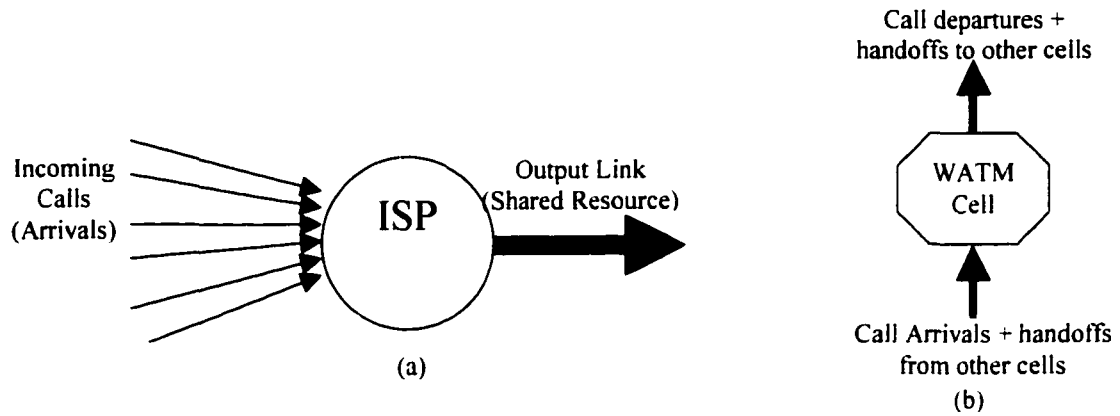
This chapter presents the simulation results of the allocation of resources algorithm presented in chapter 2. Two systems were used to simulate the proposed algorithm. The first was a single cell wireless ATM system with a number of multimedia mobiles sharing the available capacity. The purpose of this first case study is to prove the efficiency of the proposed algorithm in minimizing the call blocking rate while simultaneously maximizing the available capacity. The second one was a cellular system consisting of 3 neighboring cells. Again, each cell had a set of multimedia mobiles sharing its available capacity. Furthermore, mobiles were allowed to handoff from one serving cell to the neighboring one. The purpose of the second case study was to study the effect of mobility and handoffs on the call blocking/dropping rate, as well as the overall average QoS levels of the existing calls.

In our simulation, the video component of a multimedia call was obtained using the MPEG-1 coded movie *Star Wars* which is an example of high activity video traffic. Voice and Data substreams were generated according to an exponential distribution. The MPEG-1 coded movie generates a frame sequence of *I B B P B B P B B P B B* at a rate of 24 frames per second [33],[34]. Each call has an average duration of 5 minutes of movie time. Therefore the movie was divided into 24 different calls. The simulator was written using MATLAB. To generate three QoS levels for the video substream, we adopted the scheme described in chapter 2 where the B frames were dropped to obtain a lesser QoS level, then both the B and P frames are dropped to obtain the least video quality. Whereas, the high, medium and low QoS levels for audio were generated by varying the average bit rate of the audio source from

32 to 16 to 8 Kbps, respectively. Similarly, QoS levels for data were generated by varying the data rate from 64 to 32 to 16 Kbps.

3.1 Case Study 1

Fig. 3.1 shows a block diagram for the first system under study. It consists of a single wireless ATM cell. The cell has a capacity of 60,000 ATM cell/sec (≈25 Mbps). The algorithm can be equally applied on an internet service provider ATM system with a shared link of capacity of 60,000 ATM cell/sec. Fig.3.2 shows the output of the simulation after 2 hours of simulation time. The average call arrival rate was 1 call every 1.5 minutes. The number of calls was varied between 1 and 11, whereas the capacity was set to 60,000 ATM cells/sec. All calls were multimedia calls using all three substreams. The solid shape curve represents the bandwidth allocated to existing calls (in cells/sec) after using the proposed algorithm. The dashed curve represents the bandwidth needed by the same calls if they were granted the highest QoS level (QoS index=1). Both curves are identical as long as the available capacity is less than or equal to the required bandwidth. During periods of over-load, it is clear that the algorithm has significantly reduced the cell loss rate while optimizing usage of the available capacity.



(a) ISP with a shared output link. (b) Wireless ATM cell
Fig. 3.1 First System Under Study

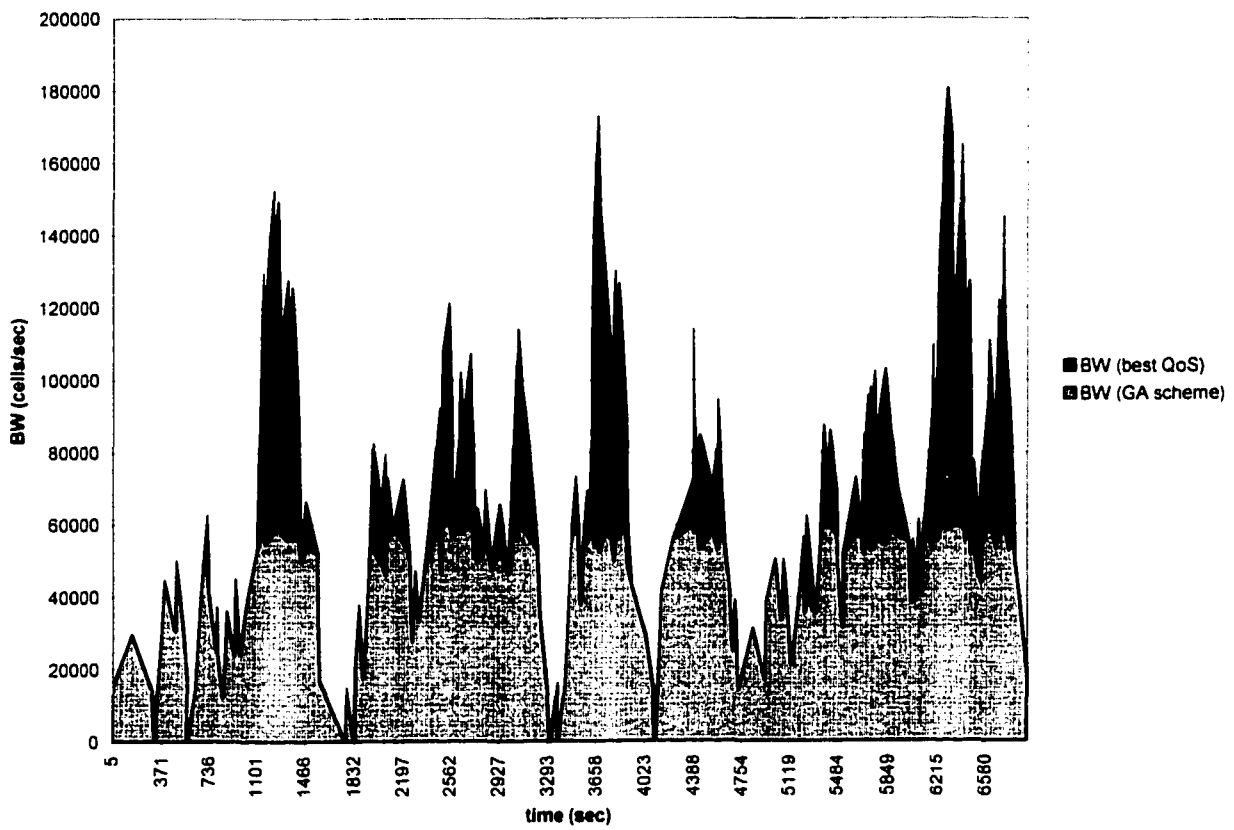


Fig. 3.2 Bandwidth Requirement versus Time

Fig. 3.3 shows the number of calls present in the system (with and without using the proposed GA algorithm) versus time. Clearly, the system was able to admit as much as 11 calls simultaneously, compared to 4 calls without control. This represents a gain of 175% in the number of admitted calls. The average percentage gain in the number of admitted calls during the whole simulation time is 55.3% whereas the maximum gain was 267%.

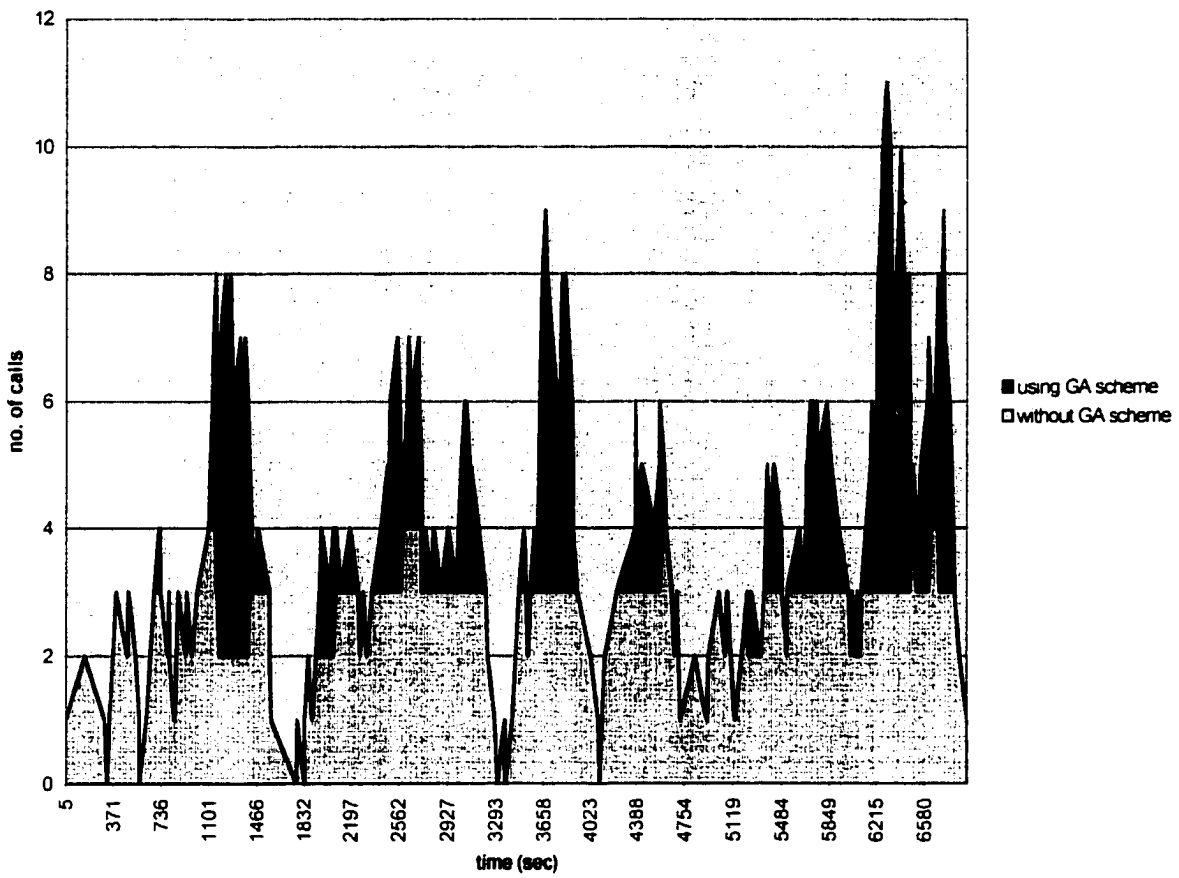


Fig.3.3 Number of calls versus time

The price to be paid for such gain is reported in Fig. 3.4. It shows a slight and graceful drop in the quality at high utilization. The minimum average QoS level that was delivered was 16.9. This means that the worst case reduction in the quality is almost 26.5% only during over-load periods. This is a minimal price to be paid compared to the advantages obtained. A QoS index of zero denotes no calls were available at that time.

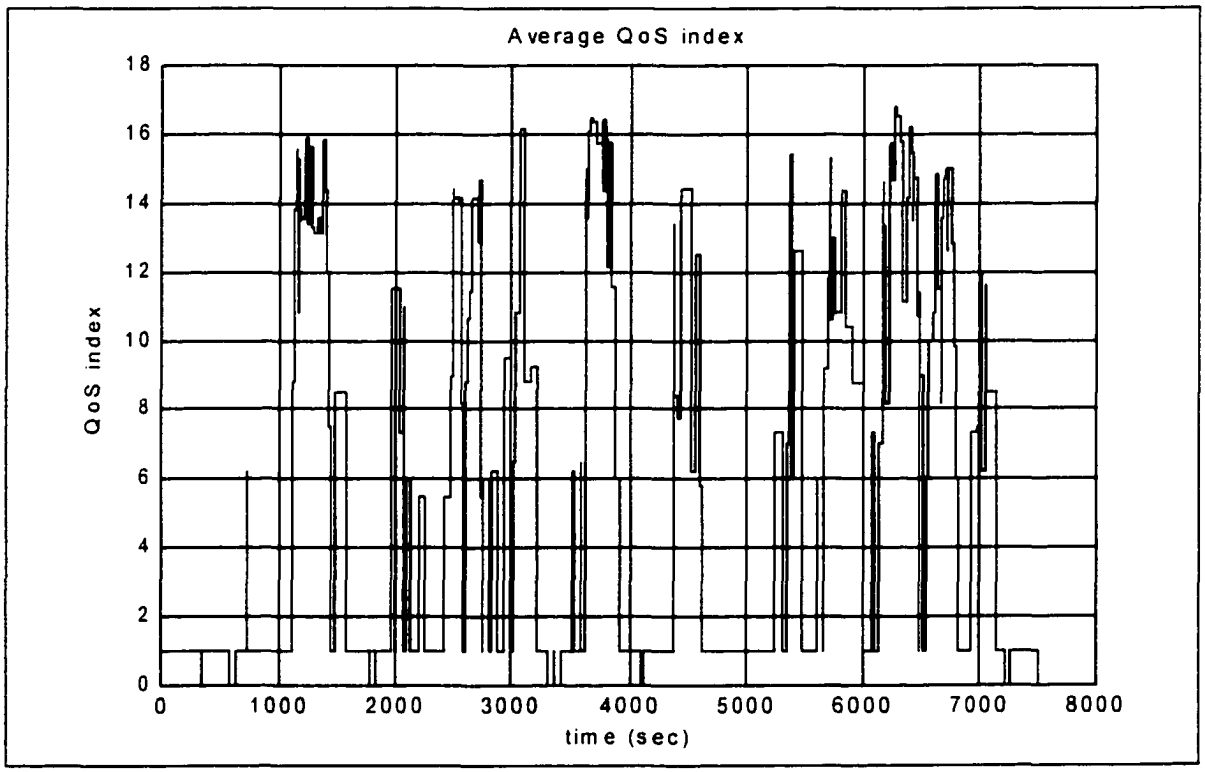


Fig. 3.4 Average QoS index versus Time

Fig. 3.5 shows the call blocking rate without the proposed allocation of resources algorithm. As shown in figure the blocking rate could be as high as 73% in some cases (when the available capacity is less than the required bandwidth). These figures illustrate the need for

such adaptive allocation of resources algorithm. When using the suggested adaptive algorithm, the system did not have to block any calls. On the contrary it admitted new calls by decreasing the QoS level of existing calls to free some bandwidth for the new arrivals.

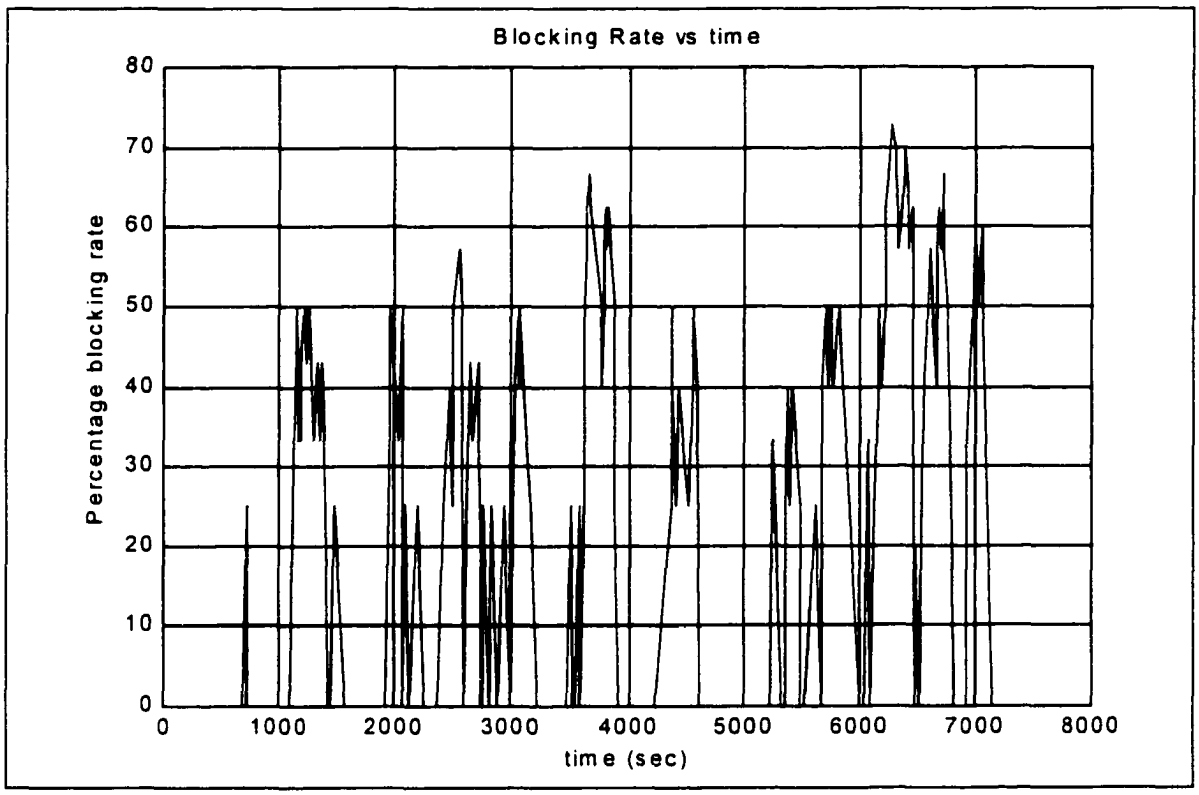


Fig. 3.5 Blocking Rate versus Time

Figure 3.6 shows a histogram for the QoS index for a sample of calls. The histogram is shown for five different calls in five different traffic conditions. The traffic loads and the corresponding number of calls are: (1) very low traffic (1 to 4 calls), (2) low traffic (5 to 6 calls), (3) moderate traffic (7 to 8 calls), (4) high traffic (9 to 10 calls) and (5) very high

traffic (more than 10 calls). Each histogram for each of these traffic loads is generated by tracing the QoS index assigned to a certain call during its lifetime in the system. It is shown that as traffic load increases, the number of times of allocations of higher QoS indices (less QoS level) starts to occur. It is also shown that the least QoS level granted was level 23 which is still very graceful to the user.

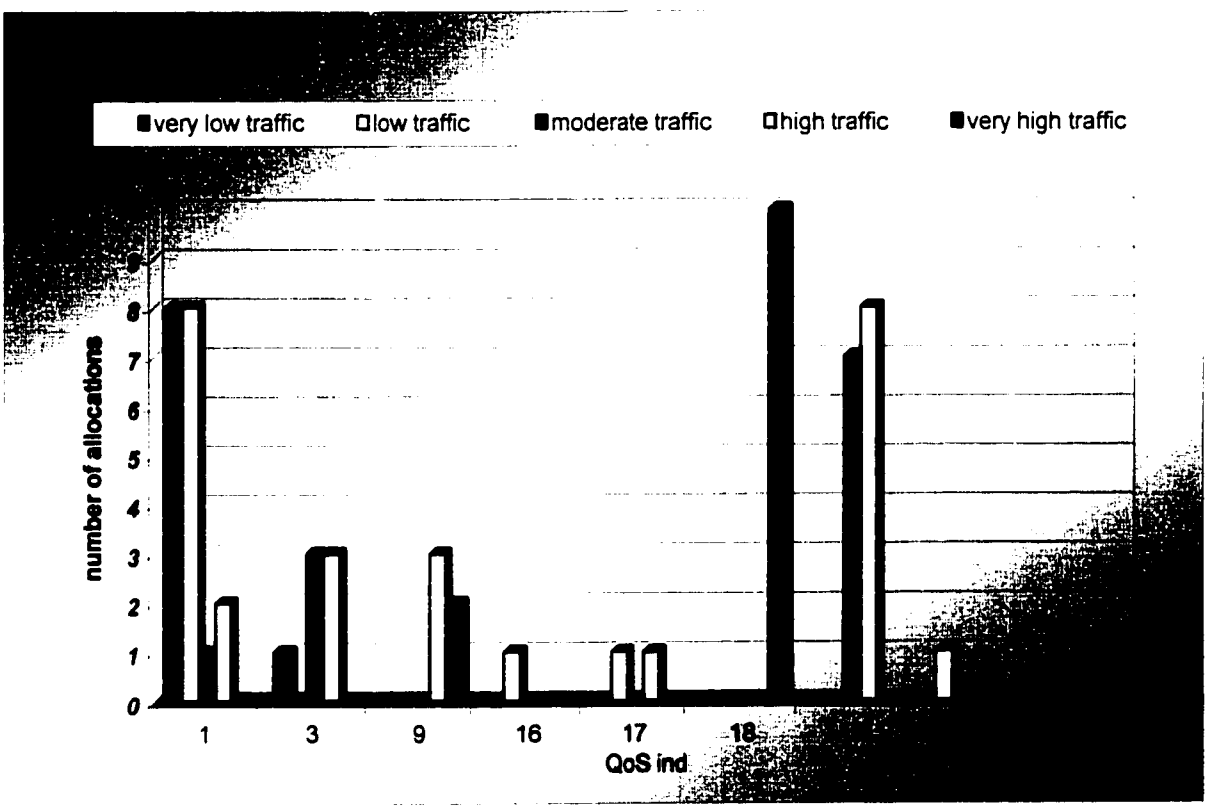


Fig. 3.6 QoS Histogram

3.2 Case Study 2

As mentioned earlier in this chapter, in the second study we aim to study the effect of the mobility and handoff occurrences on the QoS levels, call blocking rate and available capacity utilization. Fig. 3.7 shows a block diagram for the system under study. It consists of three wireless ATM cells. Each of which has a capacity of 60,000 ATM cell/sec. During handoff, traces of a certain call were kept the same after the handoff to another cell was executed.

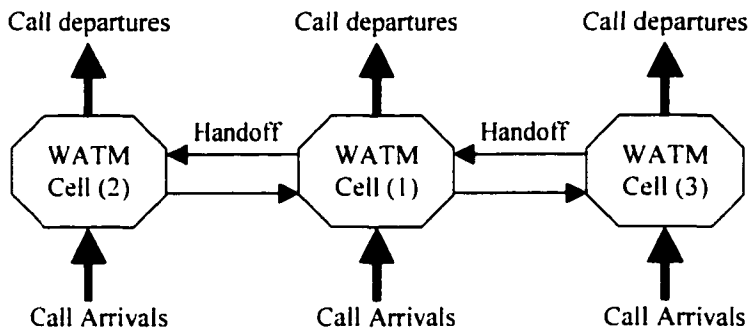


Fig. 3.7 Second System Under Study

Fig.3.8 shows the output of the simulation after 2 hours of simulation time. The average call arrival rate was 1 call per minute. The number of calls was varied between 1 and 14, whereas the capacity was set to 60,000 ATM cells/sec. The solid shape curve represents the bandwidth allocated to existing calls (in cells/sec) after using the proposed algorithm. The dashed curve represents the bandwidth needed by the same calls if they were granted the highest QoS level (QoS index=1). Both curves are identical as long as the available capacity is less than or equal to the required bandwidth. During periods of over-load, it is clear that

the algorithm has significantly reduced the cell loss rate while optimizing usage of the available capacity.

Fig. 3.9 shows the number of calls present in the system (with and without using the proposed GA algorithm) versus time. The figure shows that the system was able to admit as much as 14 calls simultaneously, compared to 4 calls without control. This represents a gain of 250% in the number of admitted calls. The average percentage gain in the number of admitted calls during the whole simulation time is 105.1% whereas the maximum gain was 500%. Fig. 3.10 shows that the system was able to admit 12 calls simultaneously in cell number 2. The maximum gain in cell 2 was 300%, whereas the average gain was 80.9%. Similar curves for cell number 3 revealed a maximum gain of 200% and an average gain of 42.5%. The price to be paid for such gain is reported in Fig. 3.11. It shows a slight and graceful drop in the quality at high utilization. The minimum average QoS level that was delivered was 19.4. This means that the worst case reduction in the quality is almost 29% only during over-load periods. A QoS index of zero denotes no calls were available at that time. Similar results are reported in Fig. 3.12 for physical cell 2. The maximum degradation in the mean QoS level cell 2 is 17.2 (26.88%). Similar curves for cell 3 revealed a maximum degradation in the mean QoS level to be 15.3 (23.9%). This is a minimal price to be paid compared to the advantages obtained.

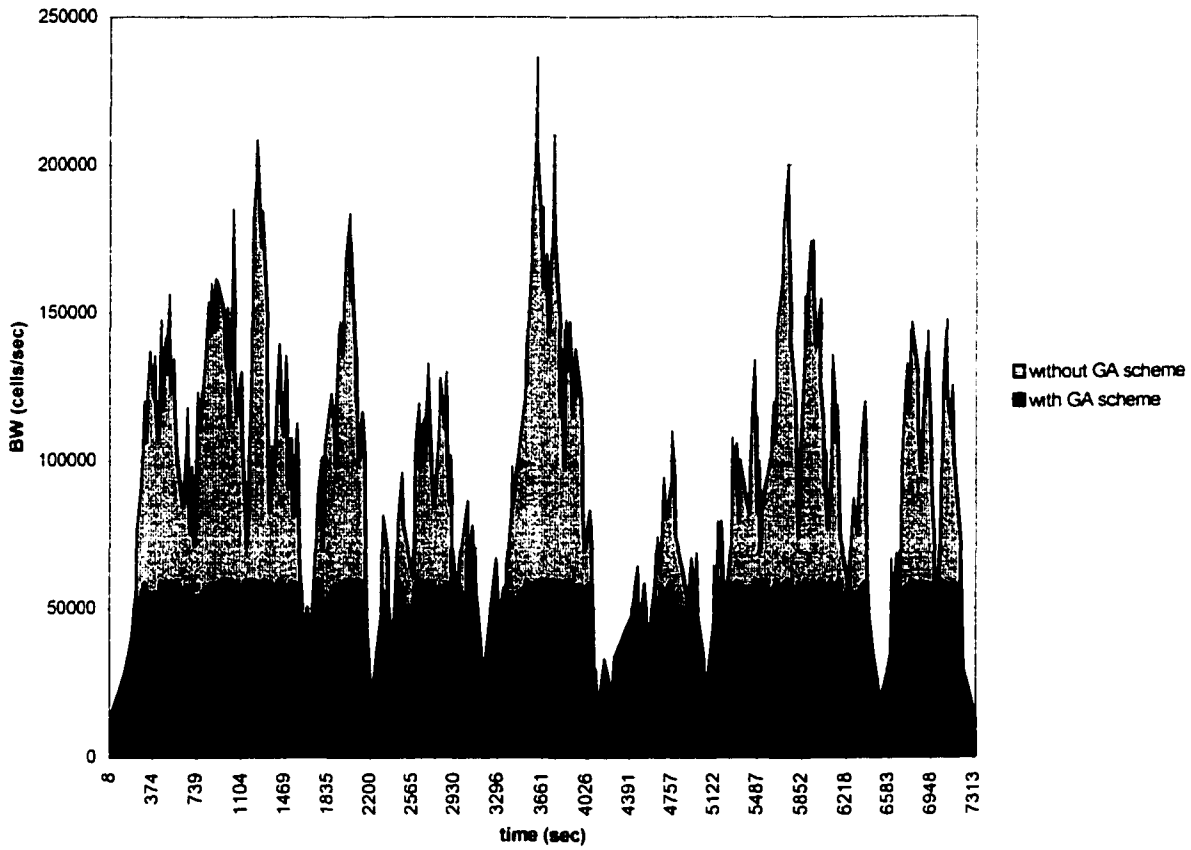


Fig.3.8 Bandwidth Requirement versus time

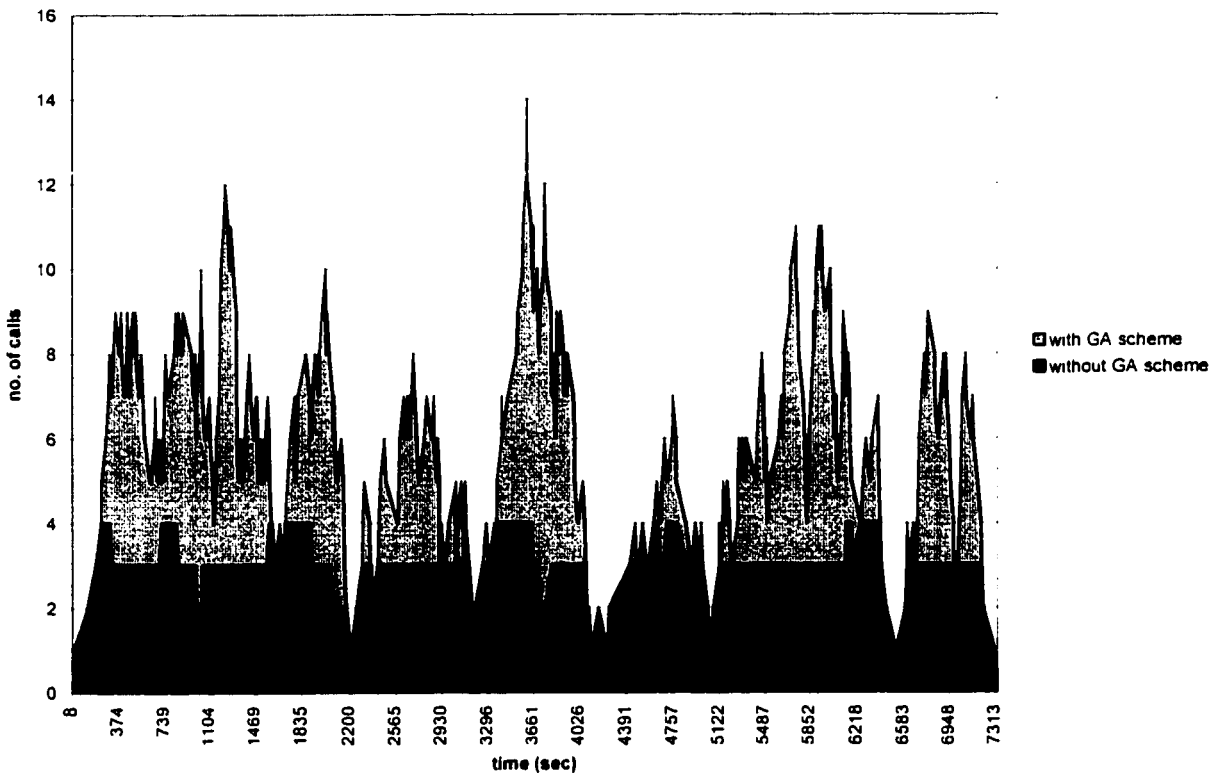


Fig.3.9 Number of calls versus time in cell 1

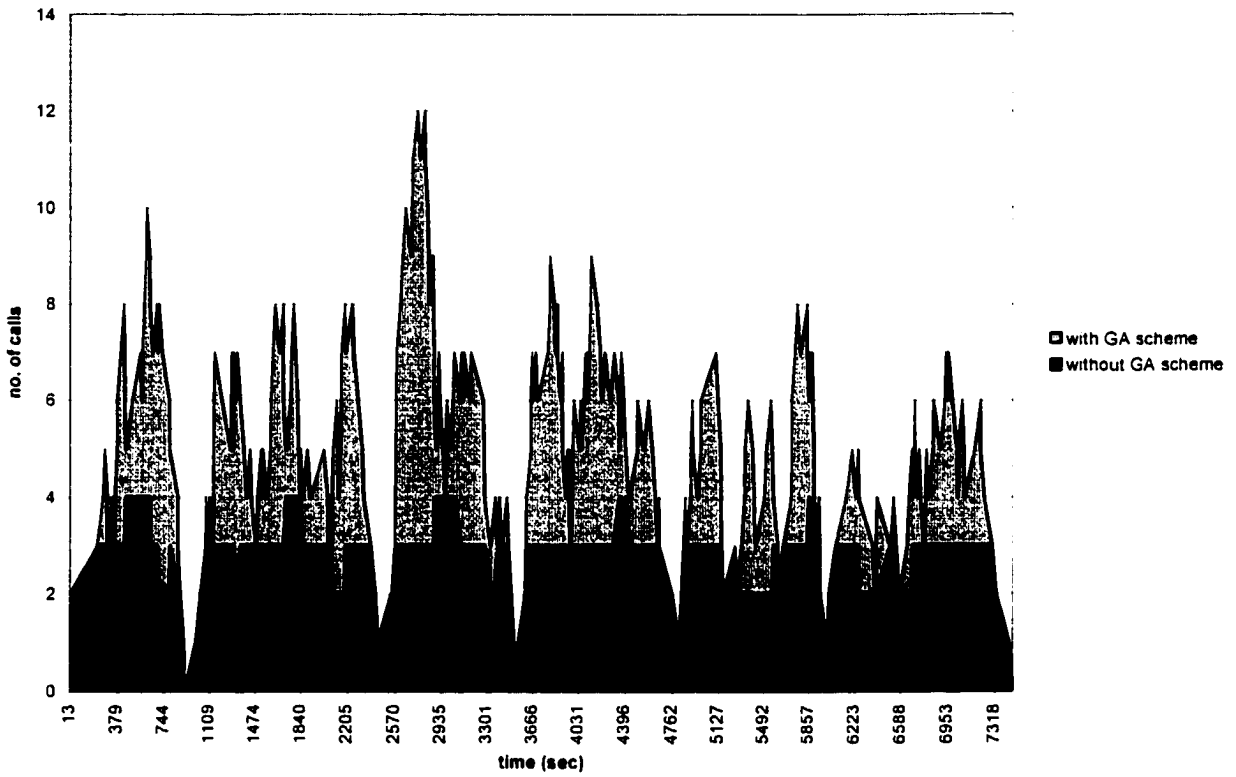
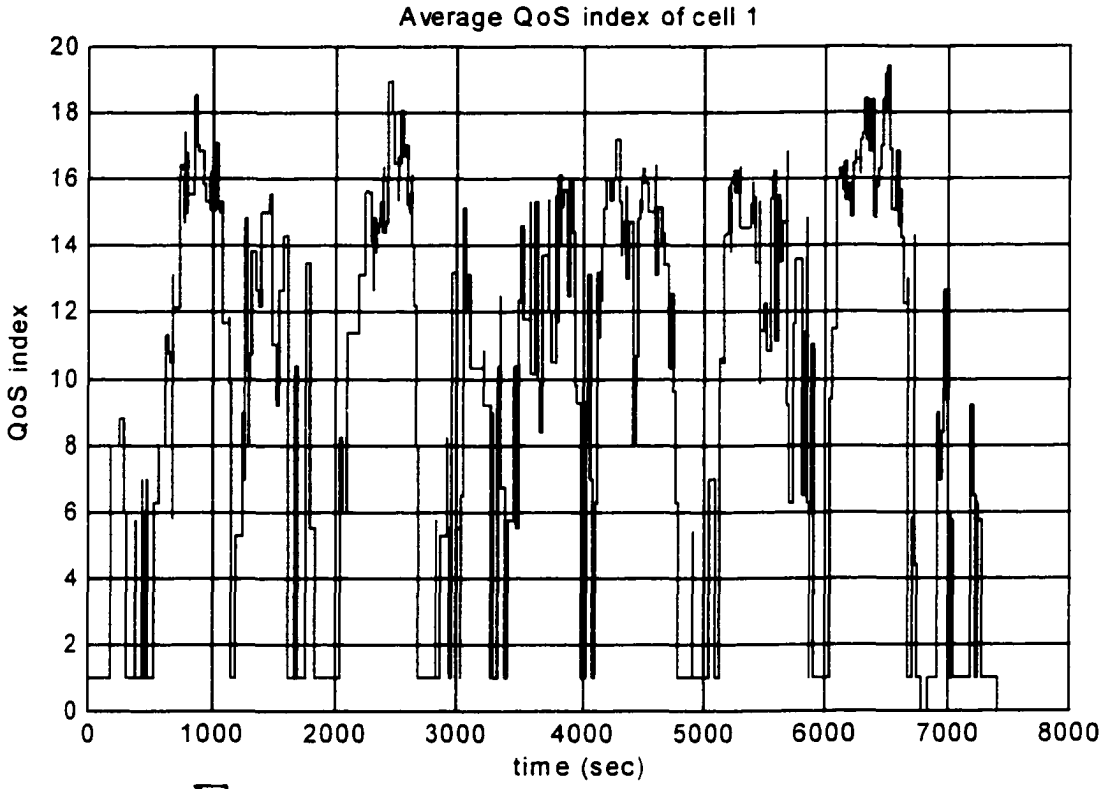


Fig.3.10 Number of calls versus time in cell 2



$$mean\,\text{QoS index} = \sum (QoS\ indices / number\ of\ calls)$$

Fig.3.11 Average QoS index versus Time for cell 1

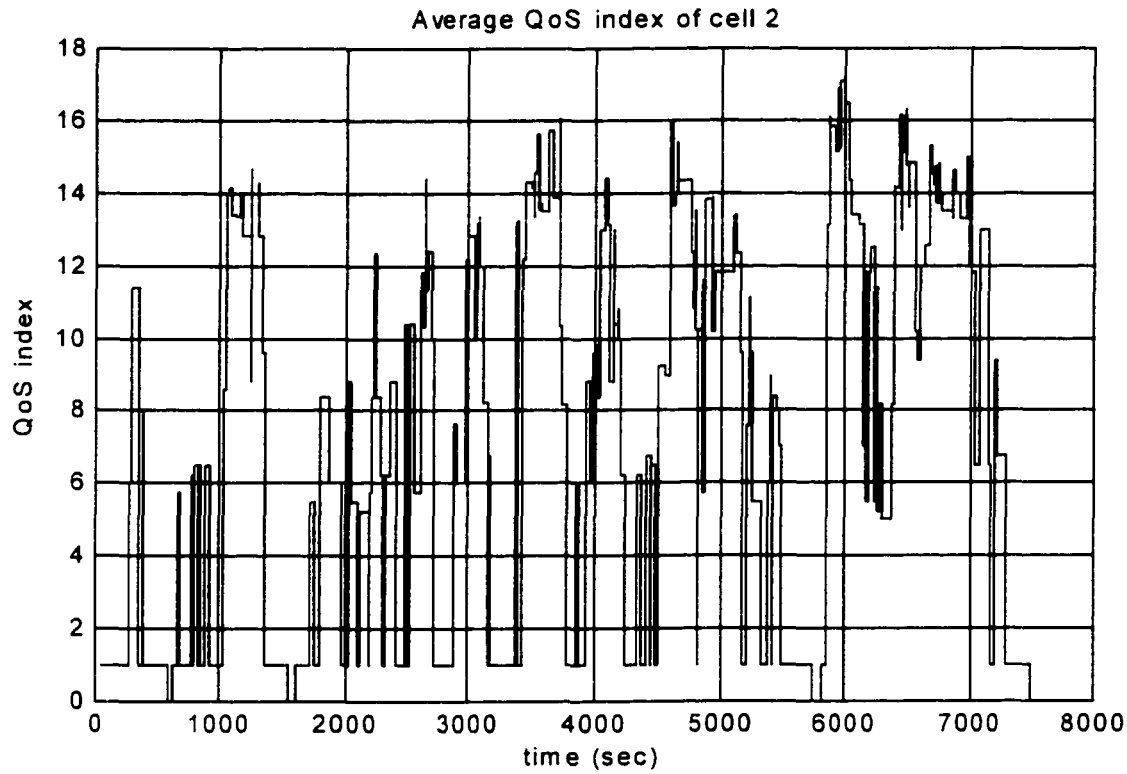


Fig.3.12 Average QoS index versus Time for cell 2

Fig. 3.13 shows the blocking rate if we do not use the proposed allocation of resources algorithm. As shown in figure the blocking rate can be as high as 77% in some cases (when the available capacity is less than the required bandwidth). These figures illustrate the need for such adaptive allocation of resources algorithm. When using the suggested adaptive algorithm, the system did not have to block any calls. On the contrary it admitted new calls by decreasing the QoS level of existing calls to free some bandwidth for the new arrivals.

Similar curves for cells 2 and 3 showed that the blocking rates if the proposed algorithm is not used were as high as 70% and 67%, respectively.

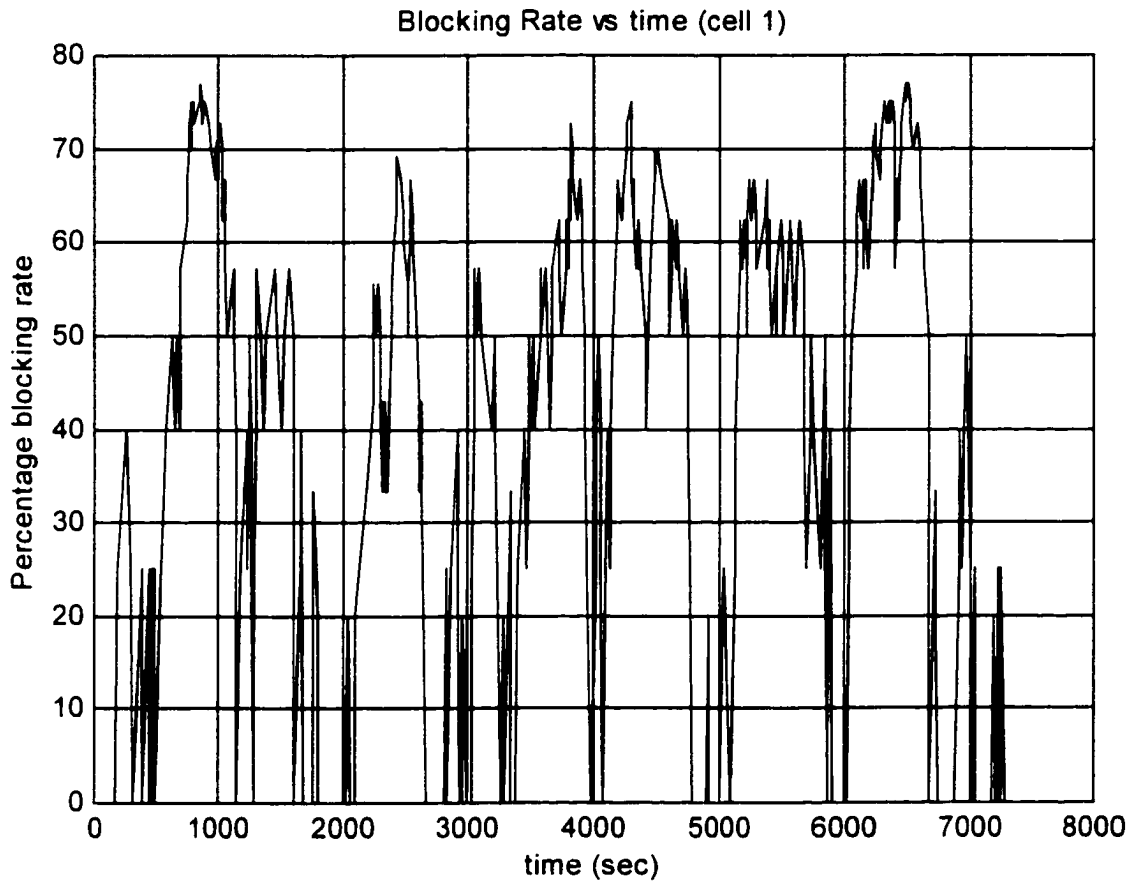


Fig.3.13 Blocking Rate versus Time for cell 1

A snapshot from the simulator output trace file shown in Fig. 3.14 shows that starting time (4), the existing calls (identified by the numbers 1,1,3 and 3) were assigned QoS indices of 19,3,2 and 8, respectively. In other words, in order not to block the newly arriving fourth call, the algorithm assigned these QoS indices to the calls. At time (5), the benefits of the algorithm are even clearer as a fifth call has been admitted. In order to do so, the following QoS indices (3,3,3,18 and 17) have been assigned to calls (1,1,3,3 and 2).

Fig. 3.15 shows the link capacity utilization versus the number of calls. The minimum utilization was 85.87%, and was as high as 99.985%. The average utilization was 97.312%. Hence, maintaining high utilization while minimizing the call blocking probability has been successfully achieved by this algorithm.

```
time(1)=3.40
QoS:     1
Calls:   1
time(2)=84.58
QoS:     1   1
Calls:   1   1
time(3)=167.88
QoS:     1   1   1
Calls:   1   1   3
time(4)=186.65
QoS:    19   3   2   8
Calls:   1   1   3   3
time(5)=259.63
QoS:     3   3   3  18  17
Calls:   1   1   3   3   2
```

Fig.3.14 Snapshot of the simulator output trace file for cell 1

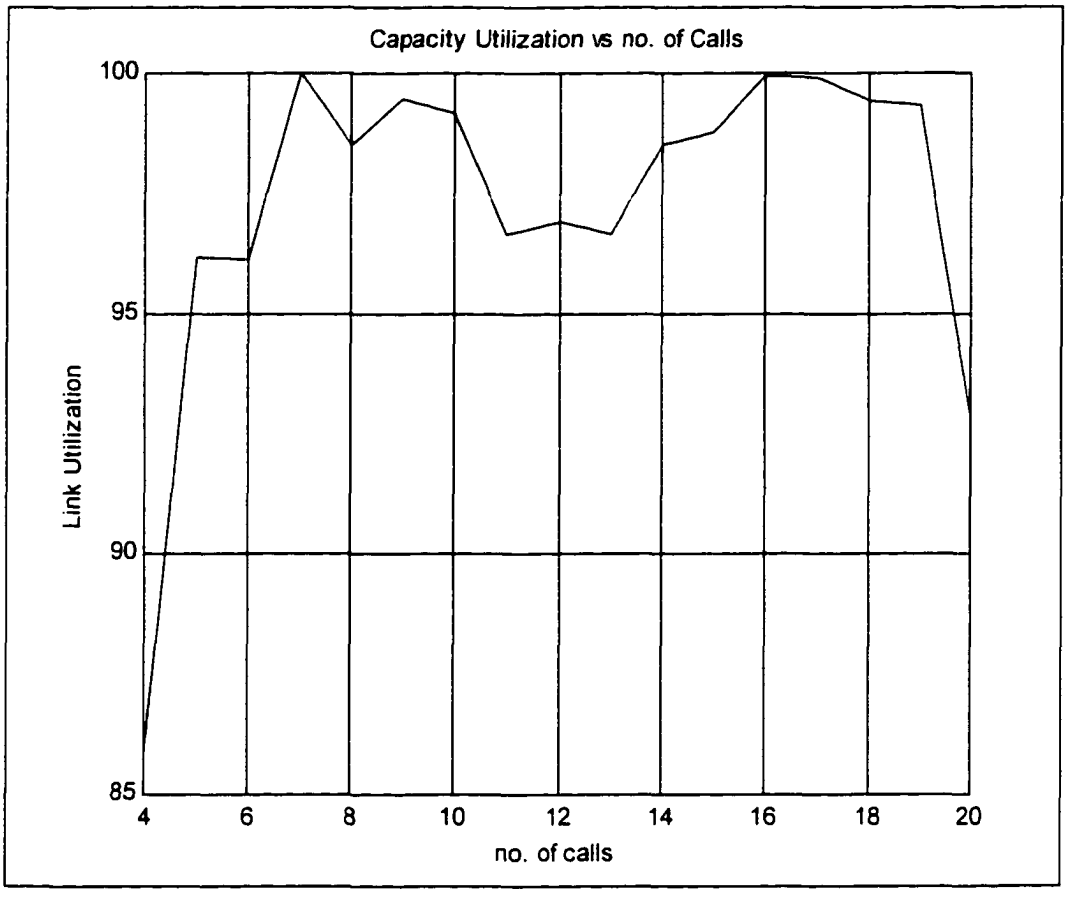


Fig.3.15 Link Utilization versus Number of Calls

Fig. 3.16 shows the effect of the handoffs on the QoS level. In our simulation, cell 1 had higher traffic load than cells 2 or 3. A positive delta QoS signifies a better QoS level being provided to the call. On the other hand, a negative delta QoS signifies degradation in the QoS after the handoff. During handoff from cell 2 to cell 1, the QoS for most calls seem to degrade. This is mainly due to the fact that cell 1 has higher traffic rate. But still the

maximum QoS degradation is a delta of 18 (a change of level from 1 to 19, i.e., 28.125% decrease in the quality).

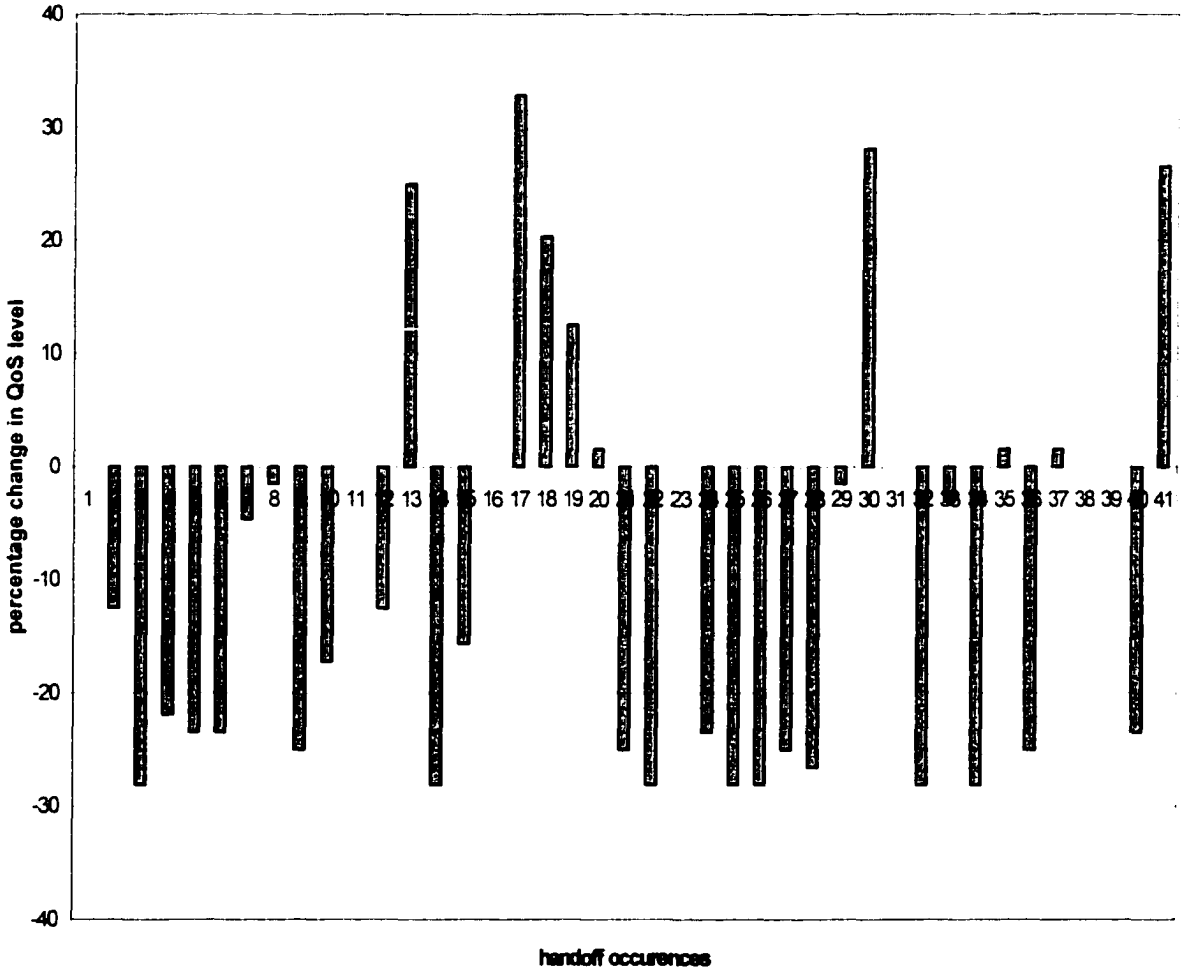


Fig.3.16 Handoff from cell 2 to cell 1

Fig. 3.17 shows the mean QoS index versus the number of calls. The figure shows that the algorithm can accommodate up to 20 calls at a price of decreasing the mean QoS index down to 28.35 (a percentage degradation of 44.3%). The curve can be used, during admission control phase, to set a threshold for the minimum quality that can be delivered by the network. For example, if an admission criteria is "admit calls as long as the mean QoS index is less than 15", then the maximum number of admitted calls will be 7.

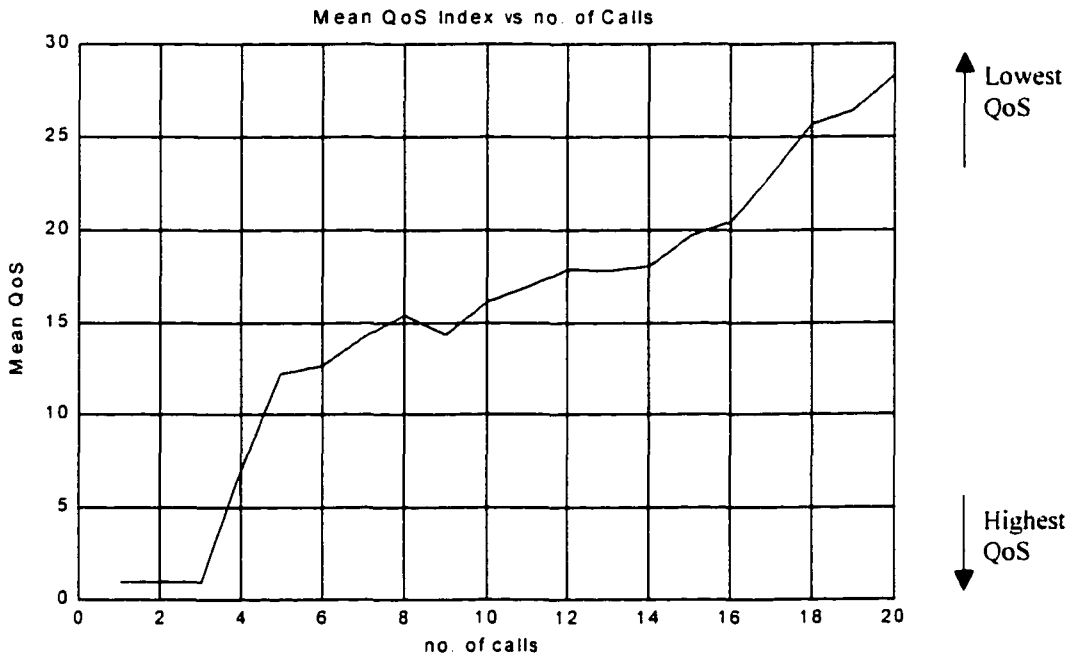


Fig. 3.17 Handoff from Cell 2 to Cell 1

The next set of curves was derived from many simulation runs. Their main objective is to show the effect of increasing the number of call arrivals on the percentage gain in number of admitted calls. Another objective is to investigate whether the available capacity has an effect on such a gain or not.

Let us define the percentage gain to be:

$$Gain(\%) = \frac{(Admitted\_calls - original\_calls)}{original\_calls} x100$$

Where

Admitted_calls represents the number of calls the system was able to admit by using the suggested algorithm;

Original_calls represents the number of calls the system admits without using the algorithm;

Fig.3.18 Shows the percentage gain versus the number of calls for two different ranges of minQ levels for a number of simulation runs. The available capacity for those simulation runs was 25 Mbps (60,000 cells/sec). The objective of such a curve is to show the percentage gain in the number of admitted calls (when using the algorithm) under different call loads and for different minQ levels. As shown, for minQ levels ranging from 17 to 31, the percentage gain saturates 360% when the number of calls reaches 16 calls. For minQ levels ranging from 32 to 47, the percentage gain saturates at 975% when the number of calls reaches 43.

If the available capacity increases to 75 Mbps, the percentage gain is shown to saturate at almost the same gain as that shown for 25Mbps. This is shown in Fig. 3.19. For minQ levels in the range from 17 to 31, the percentage gain saturates at 358% when the number of calls reaches 48. For minQ levels in the range from 32 to 47, the percentage gain saturates at 1114% when the number of calls reaches 127.

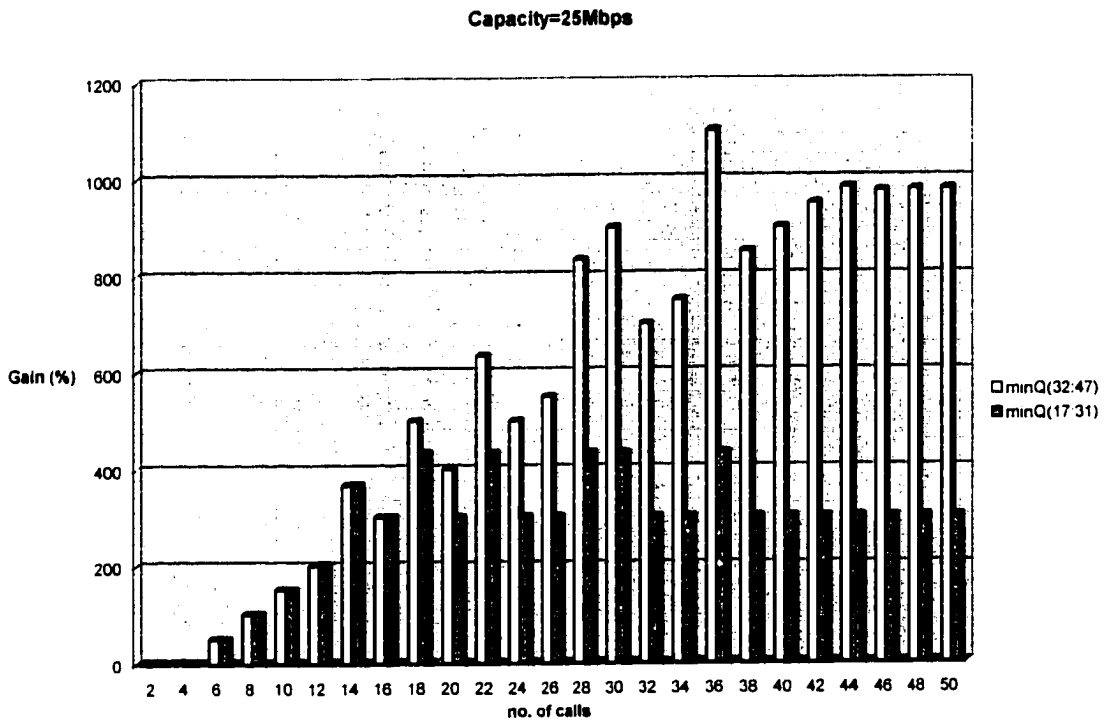


Fig. 3.18 Percentage gain versus number of calls for Cav=25Mbps

For an available capacity of 155Mbps, it is shown in Fig. 3.20 that the percentage gain saturates at 358% when the number of calls reaches 97 and at 1116% when the number of calls reaches 259 for minQ (17:31) and minQ (32:47) respectively.

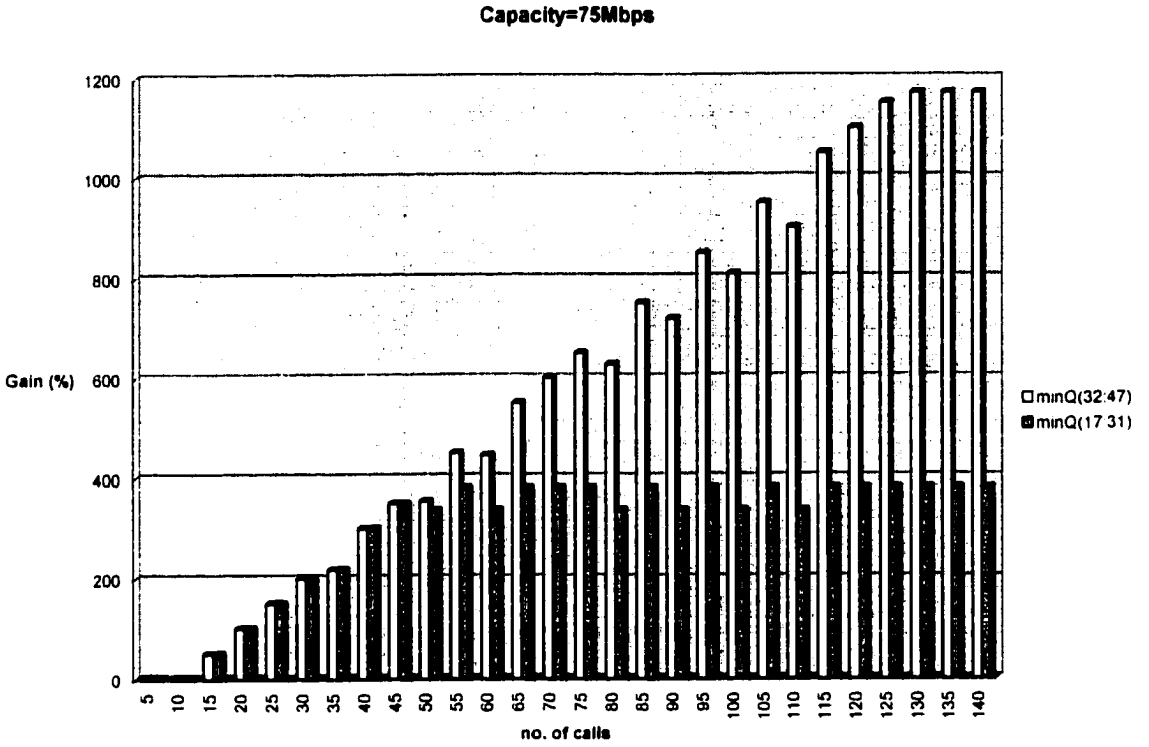


Fig. 3.19 Percentage gain versus number of calls for Cav=75Mbps

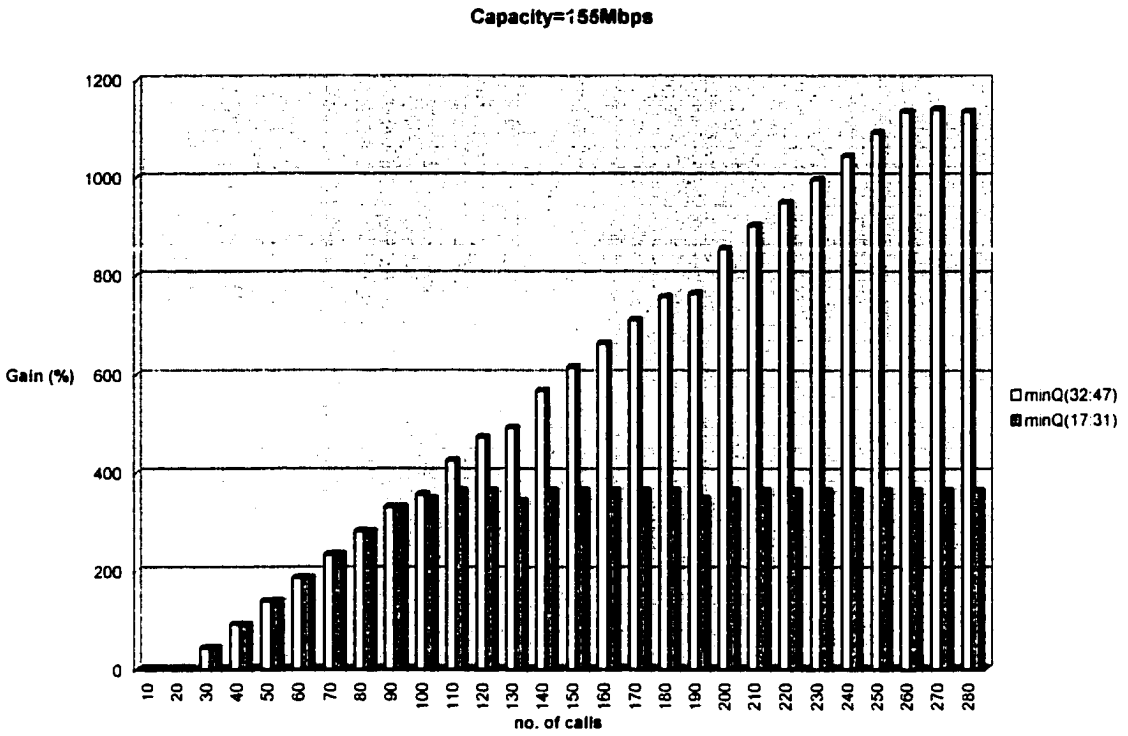


Fig. 3.20 Percentage gain versus number of calls for Cav=155Mbps

The average of many simulation runs for different available capacities ranging from 25Mbps to 155Mbps with an increment of 25Mbps is shown in Fig. 3.21. As shown in figure, the percentage gain for different capacities does not change for the same range of minQ levels. For minQ levels ranging from 17 to 31, the percentage gain saturates at 358% and for minQ levels ranging from 32 to 47, the percentage gain saturates at 1115% regardless of the available capacity.

This signifies the fact that any system deploying the proposed algorithm should be able to yield the percentage gains shown in figure at different available capacities.

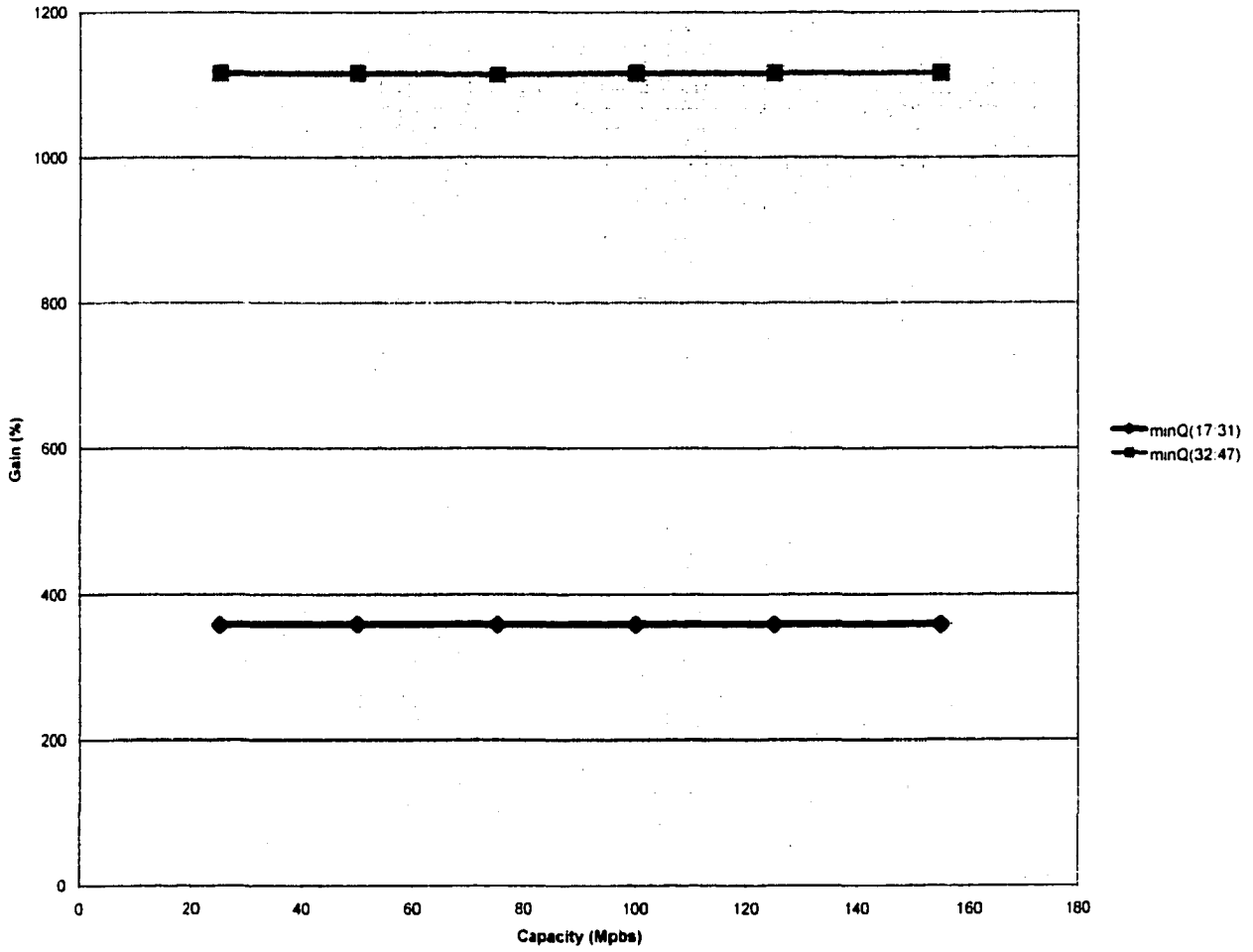


Fig. 3.21 Percentage gain versus available capacity

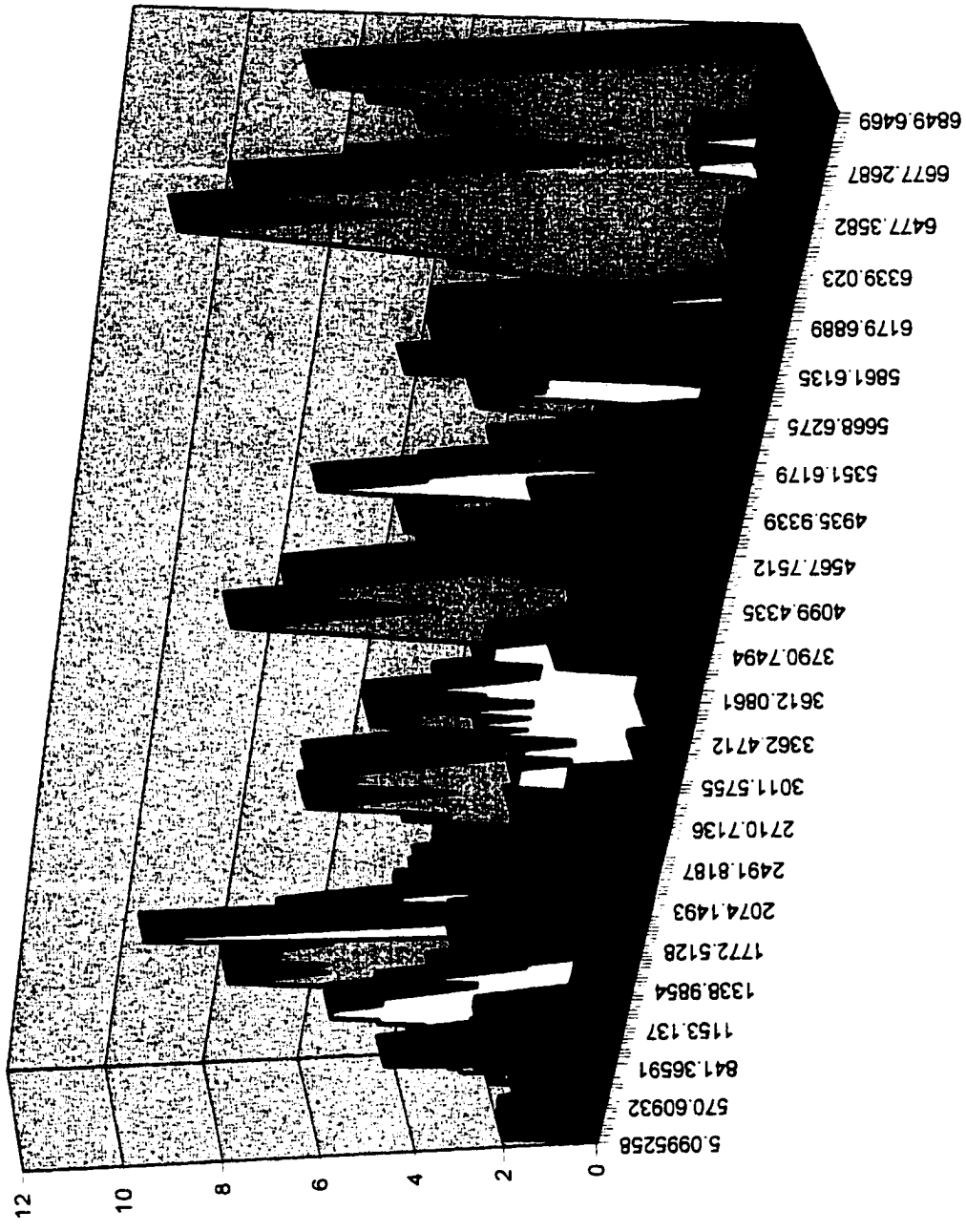


Fig. 3.22 No. of calls versus time

Fig. 3.22 shows the number of calls versus time for a number of simulation runs under different traffic load conditions. As shown, in higher traffic loads, the algorithm allowed more calls to be admitted on the expense of decreasing the QoS levels of existing calls.

Chapter 4: A QoS based Channel Borrowing Algorithm

In this chapter, we suggest a channel borrowing algorithm based on the adaptive QoS platform described in the previous chapters. In a channel borrowing algorithm, an acceptor cell that has used all its nominal channels can borrow free channels from its neighboring cells (candidate donors) to accommodate new calls. In our suggested algorithm, an acceptor cell can borrow from any neighboring (donor) cell as long as this donor cell has some channels available after satisfying the minimum QoS (minQ) level defined in the user-defined profile (UDP). A donor cell assigning QoS levels (to calls under its coverage) higher than the minQ levels defined in the UDP will declare those channels as available for borrowing by other acceptor cells. A channel can be borrowed by a cell if the borrowed channel does not interfere with existing calls. Furthermore, when a channel is borrowed, several other cells are prohibited from using it. This is called channel locking. The number of such cells depends on the cell layout and the type of initial allocation of channels to cells. In contrast to static borrowing, channel borrowing strategies deal with short-term allocation of borrowed channels to cells. Once a call is completed, the borrowed channels are returned to its nominal cell. The proposed channel borrowing algorithms differ in the way a free channel is selected from a donor cell to be borrowed by an acceptor cell.

In our suggested algorithm, the criteria for choosing the free channel include not only the number of free channels but also the QoS levels in the donor cell. The criteria is also extended to include the effect of channel locking on the number of free channels and the QoS levels on the locked cells.

4.1 Background

Throughout the description of the suggested channel borrowing algorithm, we are going to consider the hexagonal planar layout of the cells. A cell cluster is a group of identical cells in which all of the available channels (frequencies) are evenly distributed. The most widely used plan is the N=7 cell cluster [35]-[37] where the number of available channels are distributed evenly among 7 cells, which then repeats itself over and over according to Fig. 4.1. In hexagonal geometry, this reuse plan is given by

$$\frac{D}{R} = \sqrt{3N} \qquad (4.1)$$

where D is the reuse distance, R is the cell radii, and N is the modulus. We are going to consider N=7 throughout the discussion in this chapter.

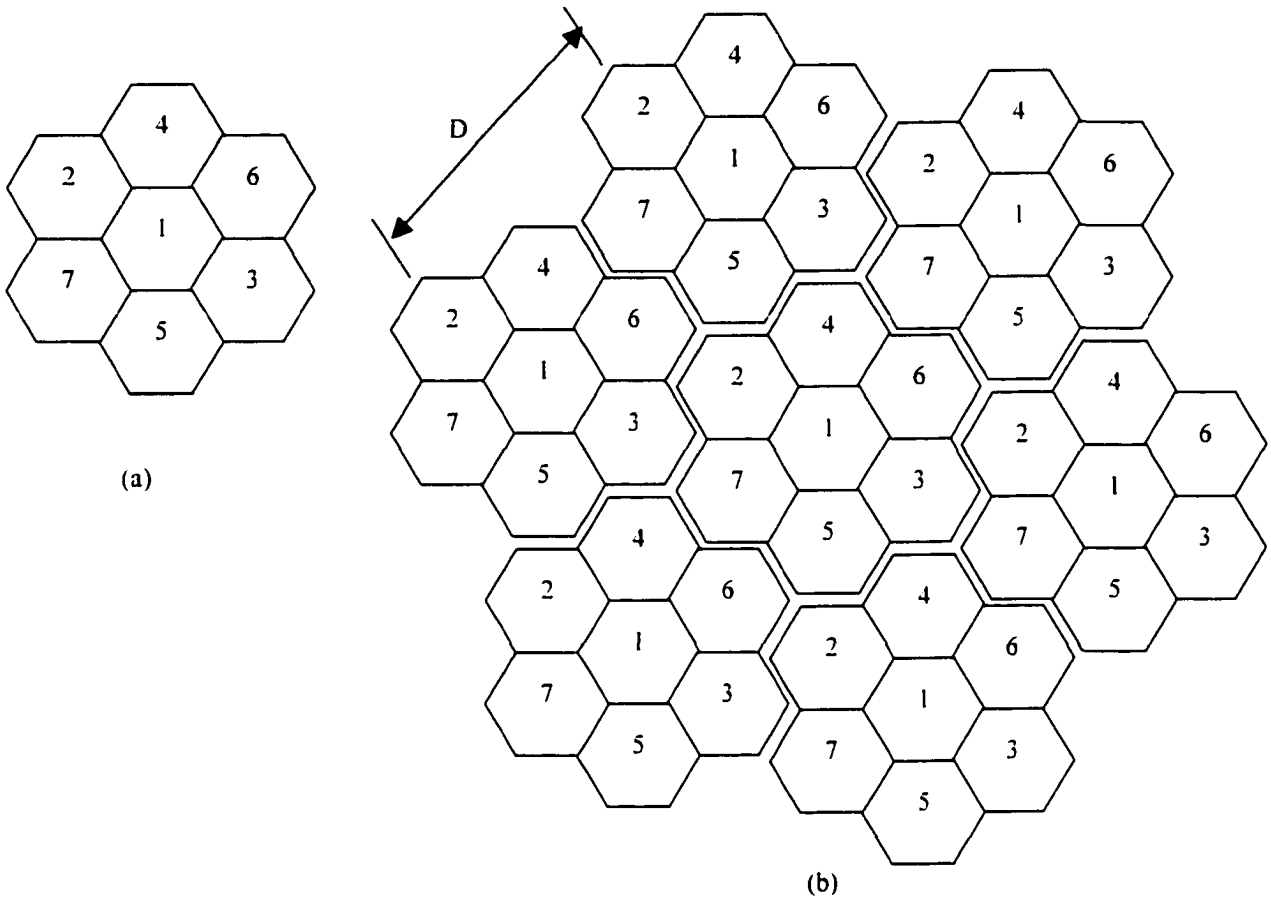


Fig. 4.1 (a) Cell cluster (N=7), (b) A seven-cell reuse plan

4.2 Related Work

The channel borrowing algorithms can be divided into simple and hybrid [38]. In simple channel borrowing algorithms, any nominal channel in a cell can be borrowed by a neighboring cell for temporary use. In hybrid channel borrowing strategies, the set of channels assigned to each cell is divided into two subsets: (1) local and (2) borrowable. The local subset is used only in the nominally assigned cell, while the borrowable subset is allowed to be lent to neighboring cells. The suggested algorithm presented in this chapter

belongs to the simple channel borrowing algorithms. A comprehensive survey of the channel borrowing algorithms is presented in [38]. A number of simple channel borrowing algorithms have been presented in the literature. *Borrow from the Richest* [39] is a simple channel borrowing algorithm that requires the acceptor cell to borrow from the cell with the greatest number of channels available for borrowing. This algorithm does not take channel locking into account when choosing a candidate channel for borrowing. *Basic Algorithm* [40] is an improved version of the *Borrow from the Richest* strategy which takes channel locking into account when selecting a candidate channel for borrowing. This algorithm tries to minimize the future call blocking probability in the cell that is most affected by the channel borrowing. This affected cell might be the donor cell or any of the cells affected due to channel locking. The number of cells affected by channel locking depends on the cell layout and the cell reuse. Instead of trying to optimize when borrowing, the *Borrow First Available* [39] algorithm selects the first candidate channel it finds. The objective is to try to minimize the complexity of the channel borrowing scheme. An Efficient *Borrowing Channel Assignment* (BCA) scheme is presented in [41]. The BCA scheme consists of two phases: (1) ordinary channel allocation phase, and (2) channel reallocation phase to improve the efficiency of the scheme.

4.3 Suggested Channel Borrowing Algorithm

As opposed to the *Borrow from the richest* [38],[39] algorithm, the suggested algorithm takes into account the effect of channel locking when choosing a candidate channel for borrowing. Furthermore, the suggested algorithm takes into account not only the number of available channels but also the average QoS level of each candidate cell. This allows the algorithm to

try to maximize the average QoS level of the calls existing in the system in addition to minimizing the call blocking probability.

Fig. 4.2 shows the block diagram of the adaptive allocation of resources algorithm when the channel borrowing algorithm is added. Channel borrowing module III is introduced to the allocation of resources algorithm. A module chooser is also introduced to decide whether the algorithm will need to execute module II or module III instead. As mentioned in earlier chapters, the algorithm is triggered whenever a call arrival or departure takes place. To adapt the algorithm for the addition of module III, some minimal changes were needed to be made in module I. Module II however is kept the same as described in earlier chapters. Once the algorithm is triggered, module I will try to assign fair bandwidth allocations to the existing calls. Fig. 4.3 shows module I after making some changes to take advantage of the channel borrowing algorithm in module III. The idea is that if any of the assigned QoS levels in module I is less than the corresponding minimum QoS level minQ, then this minQ level is assigned to the corresponding call. Then this data is sent to the module chooser for further processing. Fig. 4.4 shows the module chooser. It starts by calculating the total bandwidth needed by all existing calls if granted the QoS levels assigned by module I. If the total bandwidth exceeds the cell capacity, then module III is triggered to try to borrow some bandwidth from the neighboring cell. Otherwise, module II is triggered to try to take advantage of any bandwidth left over from module I.

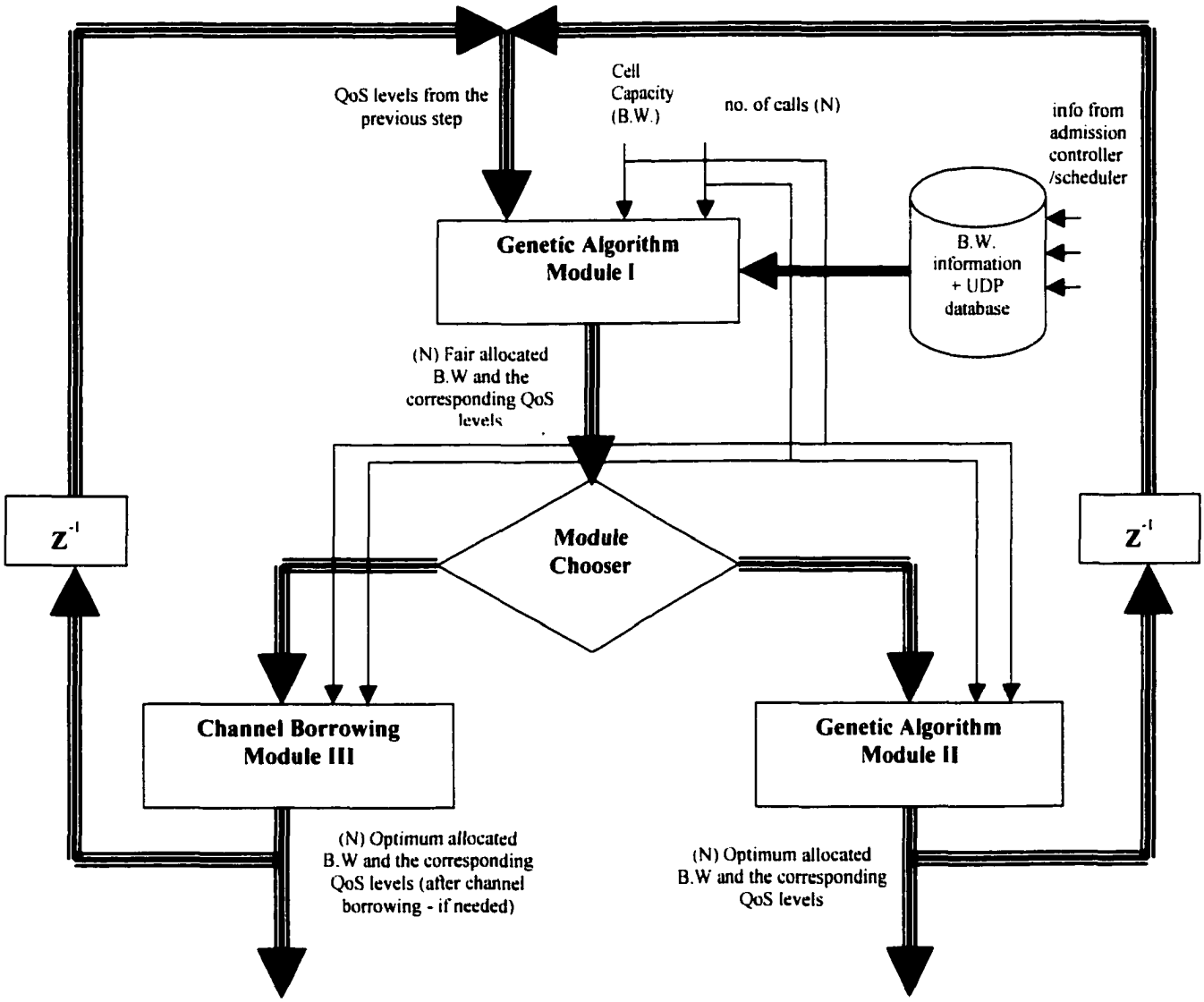


Fig. 4.2 Block Diagram of the Algorithm (after including channel borrowing)

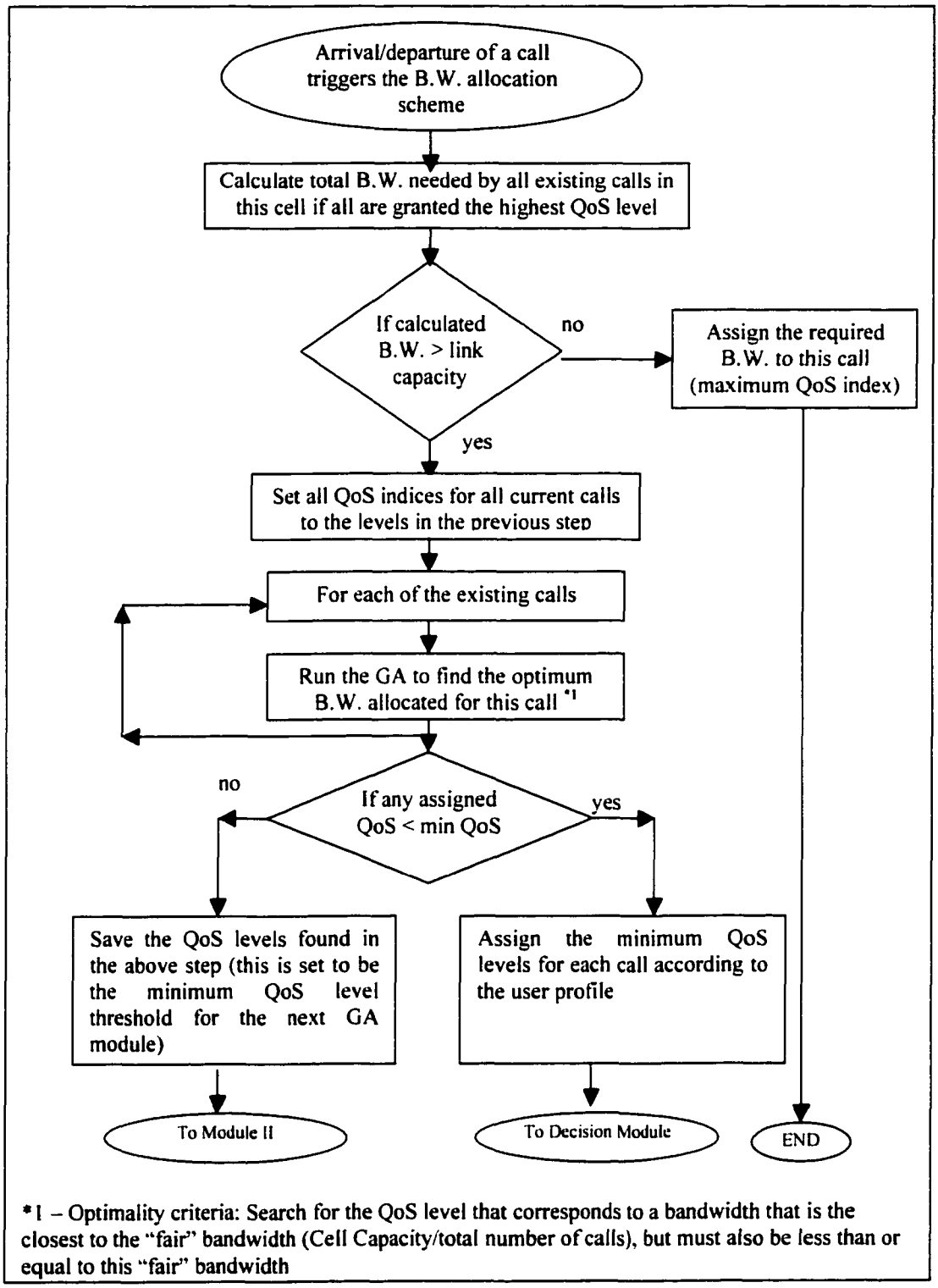


Fig. 4.3 Genetic Algorithm Module I

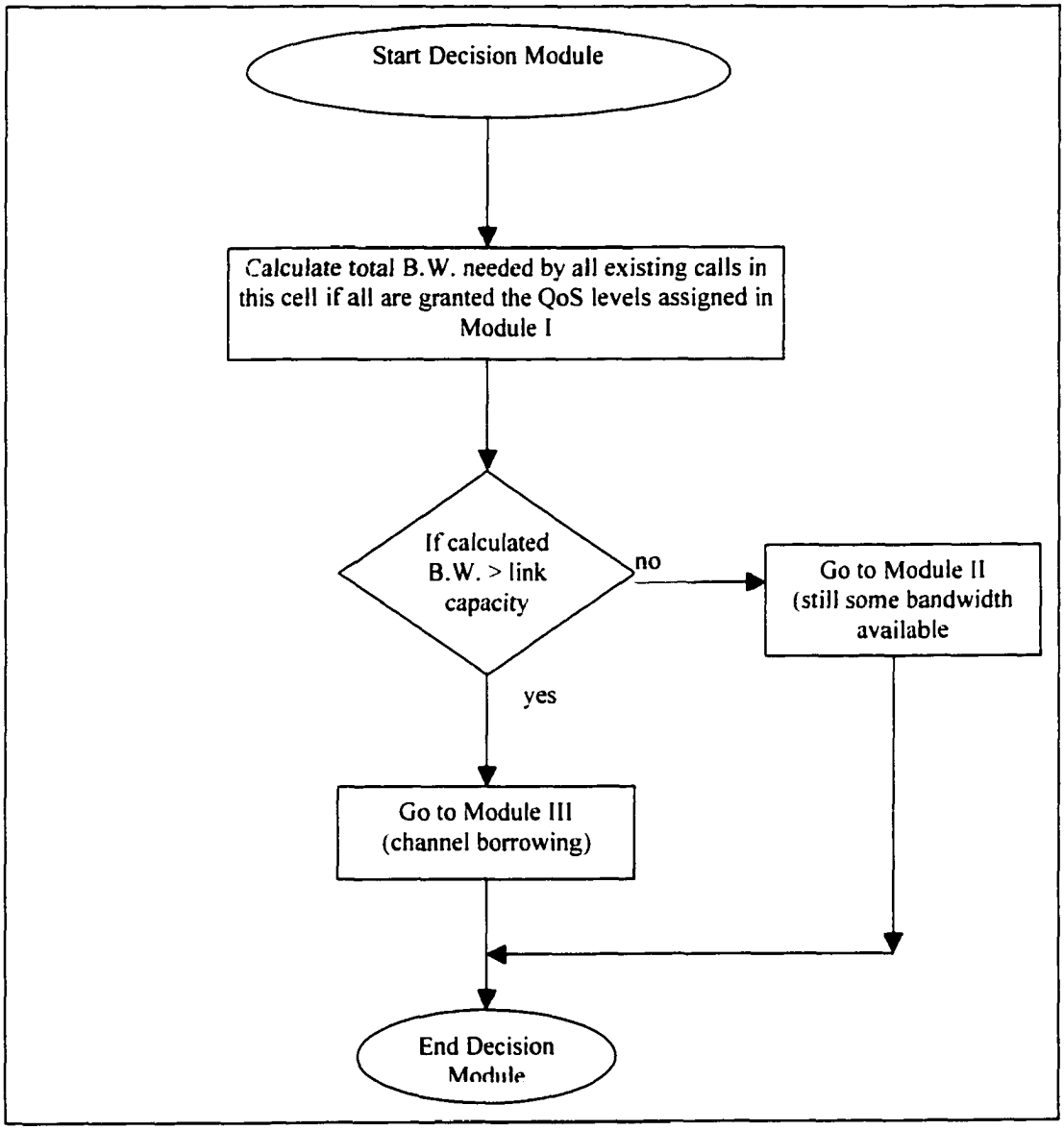


Fig. 4.4 Module Chooser

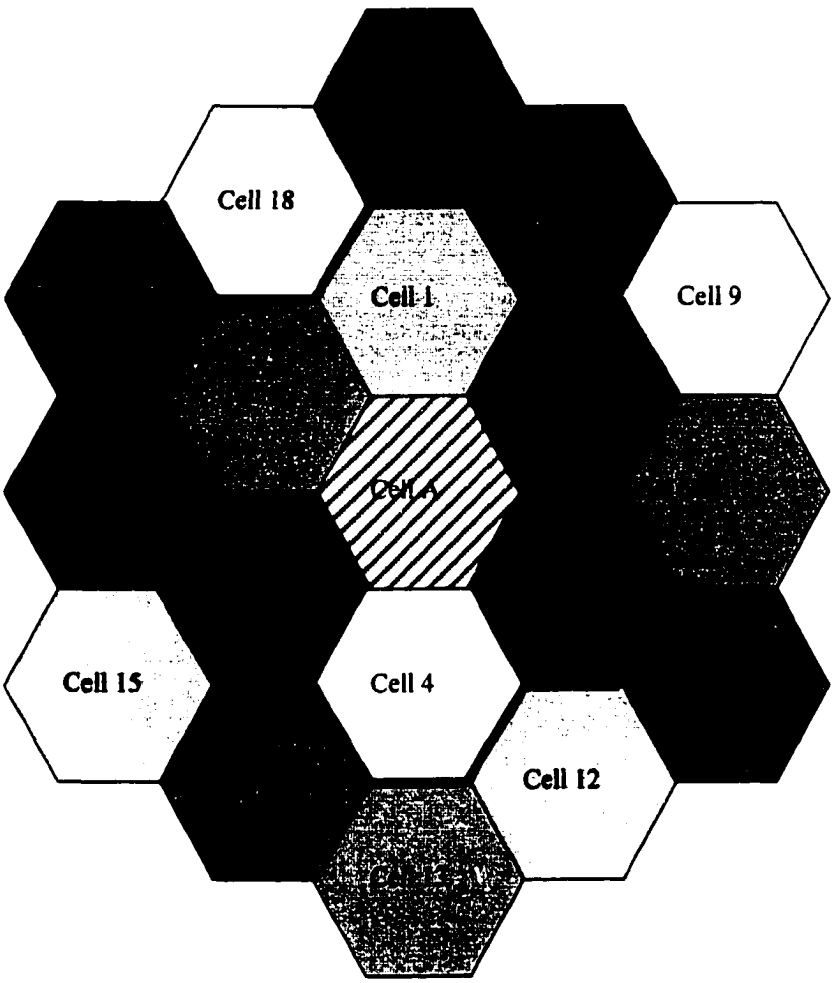


Fig. 4.5 Two Tier (N=7) Cell Layout

4.3.1 Module III

Throughout the description of module III, we are going to consider the cell layout shown in Fig. 4.5. The cell layout shown is a seven-cell cluster surrounded by a second tier of 11 cells. The total number of cells in this two-tier layout is 19 cells. The cell reuse plan is N=7. Each color in Fig. 4.5 represents a distinctive set of channel frequencies. The reason for choosing

this specific layout is that module III is going to consider the first two tiers of the neighboring cells only. Cell A is the acceptor cell. It is surrounded by two tiers of cells. Cell A can borrow from any of its first tier neighbors (cell 1, cell 2, ..., cell 6). Furthermore, it can borrow channels from one and only one neighbor (no multiple donor cells are allowed). Borrowing any number of channels from any of these 6 cells will cause these channels to be locked in another two cells. To illustrate the concept of channel locking, let us assume that cell A is going to choose to borrow a number of free channels from cell 2. This will cause these borrowed channels to be locked in both cells 14 and 17. For this cell layout (N=7), the number of cells affected by channel locking is always 2 cells. We are going to denote these 2 cells as *affected_cell1* and *affected_cell2*. In the suggested channel borrowing algorithm, each cell is required to keep track of the following parameters: (1) The number of available free channels if the existing calls are assigned their minQ levels, (2) the number of available free channels in affected_cell1 if the existing calls are assigned their minQ levels and (3) the number of available free channels in affected_cell2 if the existing calls are assigned their minQ levels. The information regarding the available number of channels in the affected_cell can be collected either periodically or on a need basis (a message sent from the cell to the affected_cell requesting immediate information regarding the number of available free channels).

Assuming that Cell A (the acceptor cell) needs N_b number of channels to borrow. The number of channels needed (N_b) should be calculated from the following equation.

$$N_b = (B_{tot} - C_{tot})/Ch_{size} \qquad (4.2)$$

Where

B_{tot} Total bandwidth needed by existing calls if assigned their minQ level.

C_{tot} Total physical cell capacity

Ch_{size} Size of each channel

Cell A will then send a *borrow_request* message to each of its first tier neighbor cells. The following routine (*Send_borrow_request*) can be run by the acceptor cell.

```
Send_borrow_request(Nb)
{
for I=1 to max_no_of_neighbors
        {
        borrow_request(cell I, Nb);
        }
}
```

Where *max_no_of_neighbors* is the number of first tier neighbors (6 cells in our case) and cell I is the cell ID.

Each of the neighboring cells receiving the *borrow_request* message will calculate the number of available channels (N_{av}) if its calls are granted their minQ level and compare it to the requested number of channels (N_b). If $N_{av} > N_b$, then the cell will send back a negative acknowledge (NACK) message informing the acceptor of the denial of its request. This is shown in Fig. 4.6 (a). Otherwise, it will send a message to each of the affected cells (due to channel locking) requesting information regarding the number of available channels (N_{av}(affected_cell)). The information regarding the available channels does not have to be gathered when the *borrow_request* message is received. As mentioned before, this

information can be gathered periodically and kept ready for any *borrow_request* message. If the returned values N_{av}(affected_cell) > N_b then this means that the requested number of channels is available for borrowing and that the locked channels (in the affected_cells) are also available. An acknowledgement (ACK) message is then sent back to the acceptor cell. Otherwise, a NACK message is sent back to the acceptor cell denoting that the locked channels will cause some calls to be dropped in the affected_cell. The following routine can be executed by the neighboring cells upon the receipt of a *borrow_request* message.

```
Receive_borrow_request(Nb)
{
calculate Nav;  /* calculate available bandwidth */
if Nav < Nb
        send(acceptor,NACK);
else {  /* there are available channels, therefore check the locked (affected) cells */
        result=ACK;
        I=0;
        while ((result==ACK) AND (I < no_of_affected_cells))
        {
                I = I+ 1;
                result=check_no_channel(affected_cell(I), Nb);
                if (result==NACK)
                        send(acceptor,NACK);  /* not enough channels in affected cell */
        }
```

```
if (result==ACK)
        {
        send(acceptor,ACK); /* Can accept the borrow_request */
        /* Now receive the Average QoS values of affected cells to send to acceptor*/
        for I=1 to no_of_affected_cells
                [average_qos(I+1),Nav(I+1)] =receive(affected_cell(I));
        /* Done with receiving all the required values */
        average_qos(1)=Calculate_qos; /* average QoS if borrow is done */
        send(acceptor,average_qos(1),Nav(1),average_qos(2),Nav(2),
        ....,average_qos(length(I),Nav(length(I) );
        }
    }
}
```

Fig. 4.6 (b) shows the flow of the messages if there are enough available channels in the neighboring (candidate) cell ($N_{av} > N_b$). If the affected cells return an ACK message, this implies that the locked channels due to the borrowing process will not cause any calls to be dropped in these cells. This is due to the fact that when the candidate cell sends a *check_no_channel* message to any of the affected cells, it will calculate the number of available channels after assigning the minQ levels to the calls existing in this cell. Therefore, immediately afterwards, the candidate cell send an ACK message to the acceptor cell informing it that it can fulfill the *borrow_request* message.

The candidate cell will then wait for each of the affected cells to send a message carrying information regarding the average QoS (*average_qos*) levels of the existing calls and the number of channels that would be available if the borrowing process is executed (N_{av}). The *average_qos* is calculated assuming the borrowing process has been executed. This will give an indication of how the borrowing process affects the QoS levels of the existing calls. In the meantime, the candidate cell will calculate the average QoS levels of its existing calls based on the same criteria. Notice that the candidate cell can calculate the average QoS level while the other affected cells are doing the same thing in parallel. Once all the information is available for the candidate cell, it will send it in a message to the acceptor cell.

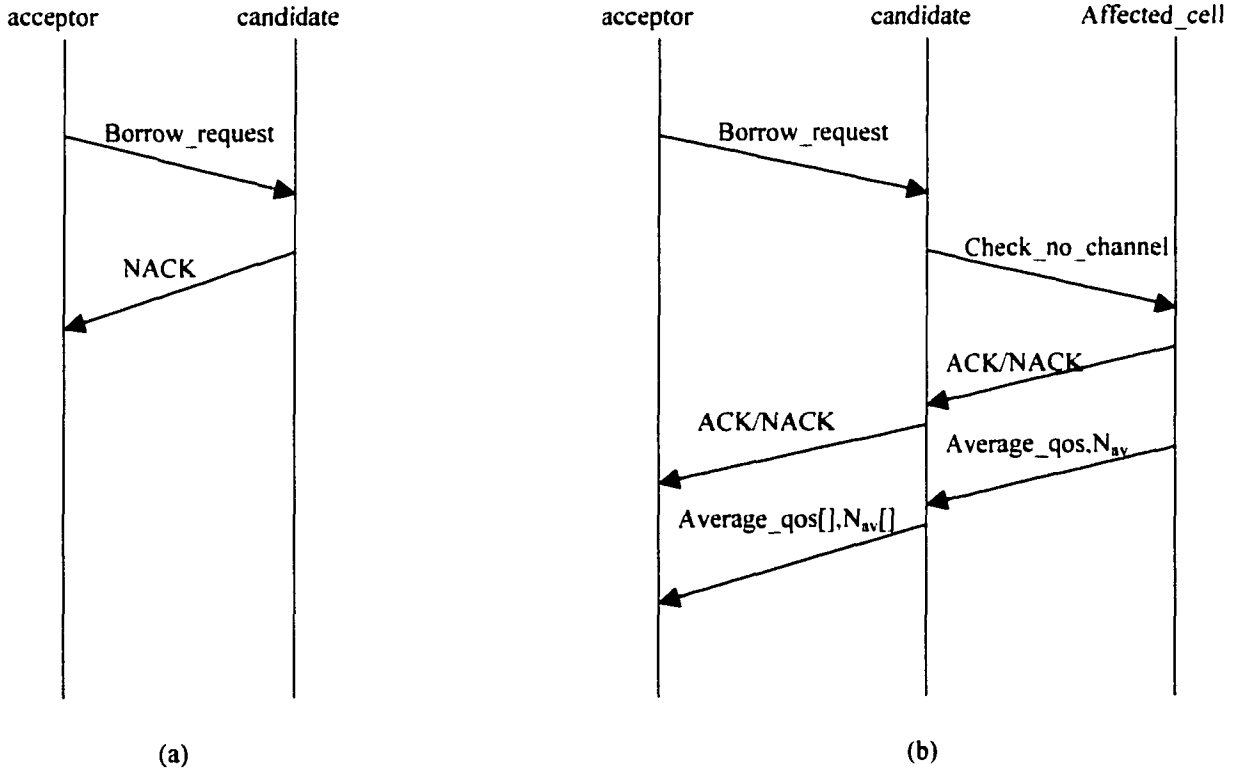


Fig. 4.6 Flow of messages between acceptor, candidate and affected cells.

(a) $N_{av} < N_b$, (b) $N_{av} > N_b$

Once all messages from all the neighboring (candidate) cells are received, the acceptor cell will start processing the information searching for the best candidate cell to become the donor. First, all cells sending back NACK messages are removed from the candidate list. The information of each of the remaining cells in the candidate list include: (1) a vector of average QoS levels of the candidate cell, affected cell 1 and affected cell 2 (*average_qos[]*), and (2) a vector of available number of channels (N_{av}).

The acceptor cell will use the following equation to calculate the cost of borrowing the requested number of channels from each cell. The cost value of each cell in the current candidate list is denoted by *borrow_cost*.

$$borrow\_cost = \left(\frac{\min(Nav)}{\max\_channels} - \frac{\min(average\_qos)}{64}\right) \qquad (4.3)$$

Where max_channels is the maximum number of channels in each of the cells. Note that the value of *average_qos* ranges from 1 to 64 (1 being the best QoS level, and 64 being the least). Therefore, the value of *borrow_cost* ranges from -1 to 1. The higher the *borrow_cost* value, the better is the candidate cell. Therefore, the acceptor cell will choose the one with the highest *borrow_cost* value.

4.4 Simulation Results

The system under study is shown in Fig. 4.7. The system consists of an acceptor cell (cell A) surrounded by two tiers of cells. The cell layout is a seven-cell cluster system (N=7). Each cell has a physical capacity of 60,000 ATM cells(packets)/sec. This is equivalent to 25 Mbps. For a 30 Kbps channels, each cell is assigned 848 channels (*max_no_channels*=848). The dotted line represents a street with high call traffic load. Along the street are cells 15, 5, A, 2 and 9. In our simulation, the traffic load in these cells is characterized by a high traffic load.

As shown in the cell layout, each of the following 6 sets represent a group of cells assigned the same set of channel frequencies: (1) cells {1,12,15}, (2) cells {2,14,17}, (3) cells {3,7,16}, (4) cells {4,9,18}, (5) cells {5,8,11}and (6) cells {6,10,13}. Therefore, the *no_of_affected_cells*=2 (i.e. when one cell is a donor cell, 2 other cells are affected due to channel locking).

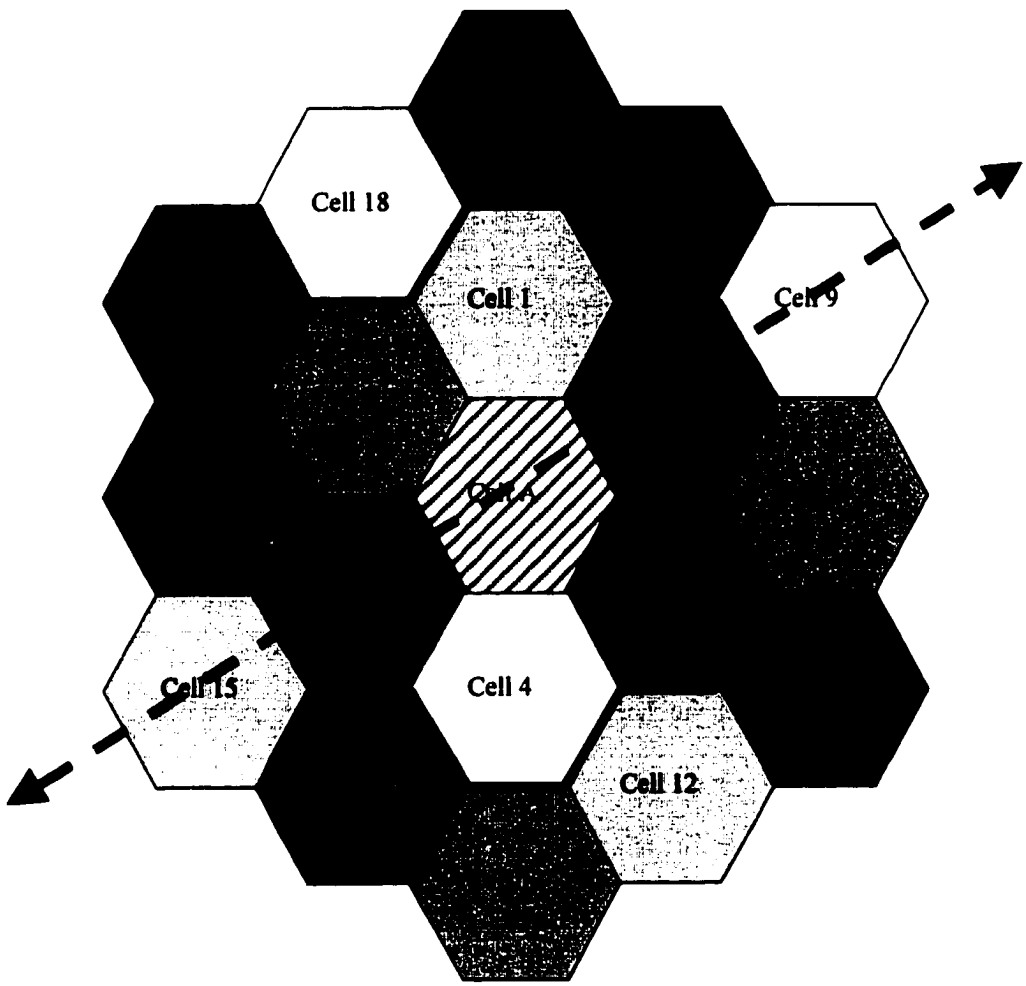


Fig. 4.7 System under study

The traffic load is multimedia traffic. In our simulation, the video component of the multimedia call was obtained using the MPEG-1 coded movie *Star Wars* as an example of high activity video traffic. Voice and Data substreams were generated according to an exponential distribution. The MPEG-1 coded movie generates a frame sequence of *I B B P B B P B B P B B*. There are 24 frames per second [33]. This set of data is described in [34]. Each call has an average duration of 5 minutes of movie time. Therefore the movie was actually divided into 24 different calls. To generate three QoS levels for the video substream, we adopted the algorithm described in chapter 2 where the B frames are dropped to obtain a lesser QoS level, then both the B and P frames are dropped to obtain the least video quality. Whereas, the high, medium and low QoS levels for audio were generated by varying the average bit rate of the audio source from 32 to 16 to 8 Kbps, respectively. Similarly, QoS levels for data were generated by varying the data rate from 64 to 32 to 16 Kbps.

The calls statistics were as follows: (a) 33% of the calls had a minQ level equals to 1, and (b) the rest of the calls had a minQ level equals to 18. The initial distribution of the calls is shown in Fig. 4.8 (the number between brackets represent the number of calls).

Assuming that a 9^{th} call needs to be admitted to cell A and that there is not enough number of free channels to satisfy the minQ level of such a call. This will trigger module III to try to borrow a number of channels from any of the 6 neighboring cells (cells 1,2,...,6). Table 4.1 shows all the information gathered by the simulator after running the algorithm including module III. The information gathered include: (1) the average QoS level of each candidate cell, affected cell 1, and affected cell 2 (*average_qos(1), average_qos(2), average_qos(3)*), (2) the number of available channels of each candidate cell, affected cell 1 and affected cell 2 ($N_{av}(1)$, $N_{av}(2)$, $N_{av}(3)$) and (3) the *borrow_cost* value of each candidate cell.

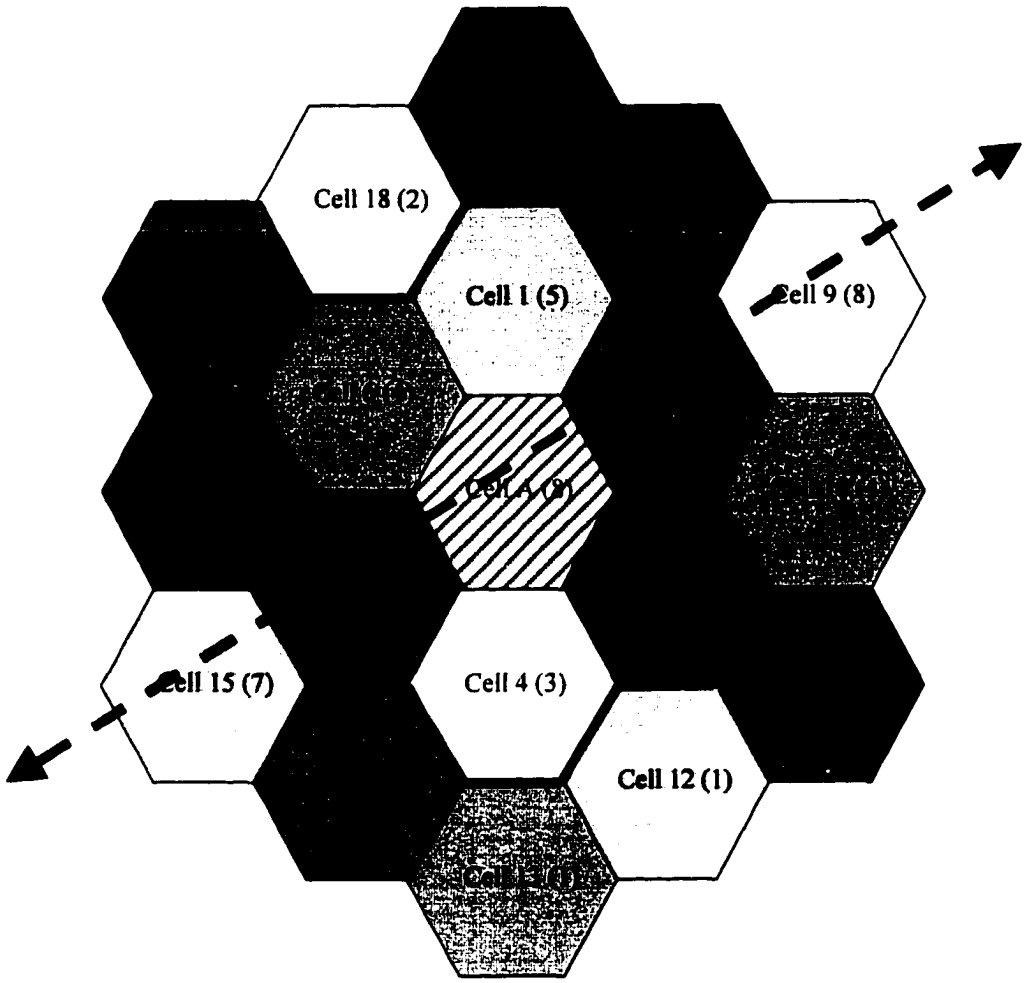


Fig. 4.8 Initial call distribution

As shown in Table 4.1, 2 of the candidate cells returned a NACK message. Cell 2 returned a NACK because there are no channels available (*Nav(1)=0*) in the candidate cell itself. On the other hand, cell 4 returned a NACK because one of the affected cell (cell 9) does not any channels available (*Nav(2)=0*) and channel locking would cause one or more calls to be

dropped. Therefore, the *borrow_cost* value is calculated for only 4 cells. Cell 6 has the higher cost value and therefore was chosen to be the donor cell.

Table 4.2 shows the same information for a higher traffic call load. As shown, the cell with the highest cost value was cell 6. The main advantage of any channel borrowing algorithm is to reduce the call blocking rate. To study the effect of the suggested channel borrowing algorithm on the system under study, we had to calculate the call blocking rate at different values of traffic load. Furthermore, our suggested algorithm not only tries to reduce the call blocking rate, but also enhances the QoS offered o the existing calls. This is mainly due to the existence of a preset range of acceptable QoS levels instead of one. Therefore, the algorithm will always try to take advantage of any free channels available to increase the offered QoS level. Fig. 4.9 shows the percentage call blocking rate of the system under study versus the percentage offered load. Using the suggested channel borrowing algorithm reduced the call blocking rate of the system. This takes place under an offered load between 40% to 80%. Below 40%, there is no need for using the channel borrowing algorithm as there are enough channel in the system to satisfy at least the minQ levels of the existing calls.

Table 4.1 Module III simulator information (initial call distribution)

Cap=60000 cell/sec = 848 channels (channel=30 Kbps).

	Cell A	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6
No. of calls	9	5	7	3	3	7	4
Calls Ids	1,2,3,4,5,6,7,8,9	8,9,10,11,12	13,14,15,16,17,18,19	19,20,21	22,23,24	1,2,3,4,5,6,7	8,9,10,11
MinQ	1,1,1,18,18,18,18,18,18	1,18,18,18,18	1,1,18,18,18,18,18	1,18,18	1,18,18	1,1,18,18,18,18,18	1,18,18,18
$N_{av}(1)$	997	471	0	502	426	178	515
ACK	needed bw=10478 / 149	Y	N	Y	N	Y	Y
Average_qos(1)	12.4	13.4	13.2	12.4	12.4	13.2	13.8
Affected Cell 1		12 (1 call)	14 (4 calls)	7 (2 calls)	9 (8 calls)	8 (4 calls)	10 (4 calls)
$N_{av}(2)$		645		405	0	515	515
Average_qos(2)		1.0		1.0		13.8	13.8
Affected Cell 2		15 (7 calls)	17 (2 calls)	16 (5 calls)	18 (2 calls)	11 (2 calls)	13 (1 call)
$N_{av}(3)$		178		471		405	645
Average_qos(3)		13.2		13.4		1.0	1.0
Borrow_cost		$0.531 * 10^{-3}$	NACK	0.2682	NACK	$-5.719 * 10^{-3}$	0.3917

Cell 6 is chosen as the donor cell (highest *borrow_cost* value).

Table 4.2 Module III simulator information (higher traffic load)

Cap=60000 cell/sec = 848 channels (channel=30 Kbps).

	Cell A	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6
No. of calls	9	6	8	4	4	8	5
Calls Ids	1,2,3,4,5,6,7,8,9	8,9,10,11,12,13	1,2,3,4,5,6,7,8	8,9,10,11	8,9,10,11	1,2,3,4,5,6,7	8,9,10,11,12
MinQ	1,1,1,18,18,18,18,18,18	1,1,18,18,18,18	1,1,18,18,18,18,18,18	1,18,18,18	1,18,18,18	1,1,18,18,18,18,18	1,18,18,18,18
$N_{av}(1)$	70478 / 997	0 / 0	0 / 0	36503 / 515	36503 / 515	0 / 0	33357 / 471
ACK	needed bw=10478 / 149	N	N	Y	N	N	Y
Average_qos(1)	12.4			13.8	13.8		13.4
Affected Cell 1		12 (1 call)	14 (4 calls)	7 (2 calls)	9 (8 calls)	8 (4 calls)	10 (4 calls)
$N_{av}(2)$		45707 / 645		28698 / 405	0 / 0	36503 / 515	36503 / 515
Average_qos(2)		1.0		1.0		13.8	13.8
Affected Cell 2		15 (7 calls)	17 (2 calls)	16 (5 calls)	18 (2 calls)	11 (2 calls)	13 (1 call)
$N_{av}(3)$		12623 / 178		33357 / 471		28698 / 405	45707 / 645
Average_qos(3)		13.2		13.4		1.0	1.0
Borrow_cost		NACK	NACK	0.2620	NACK	NACK	0.3398

Cell 6 is chosen as the donor cell (highest *borrow_cost* value).

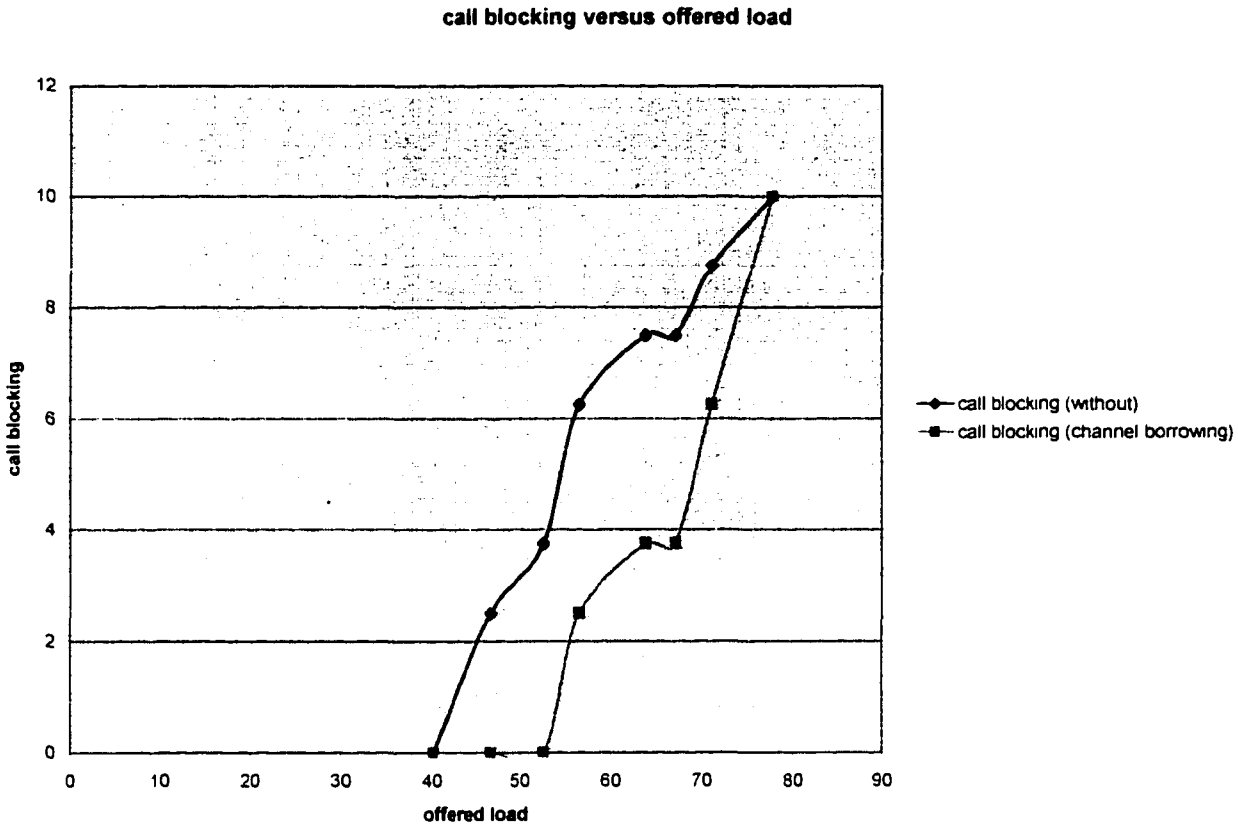


Fig. 4.9 Call blocking versus offered load

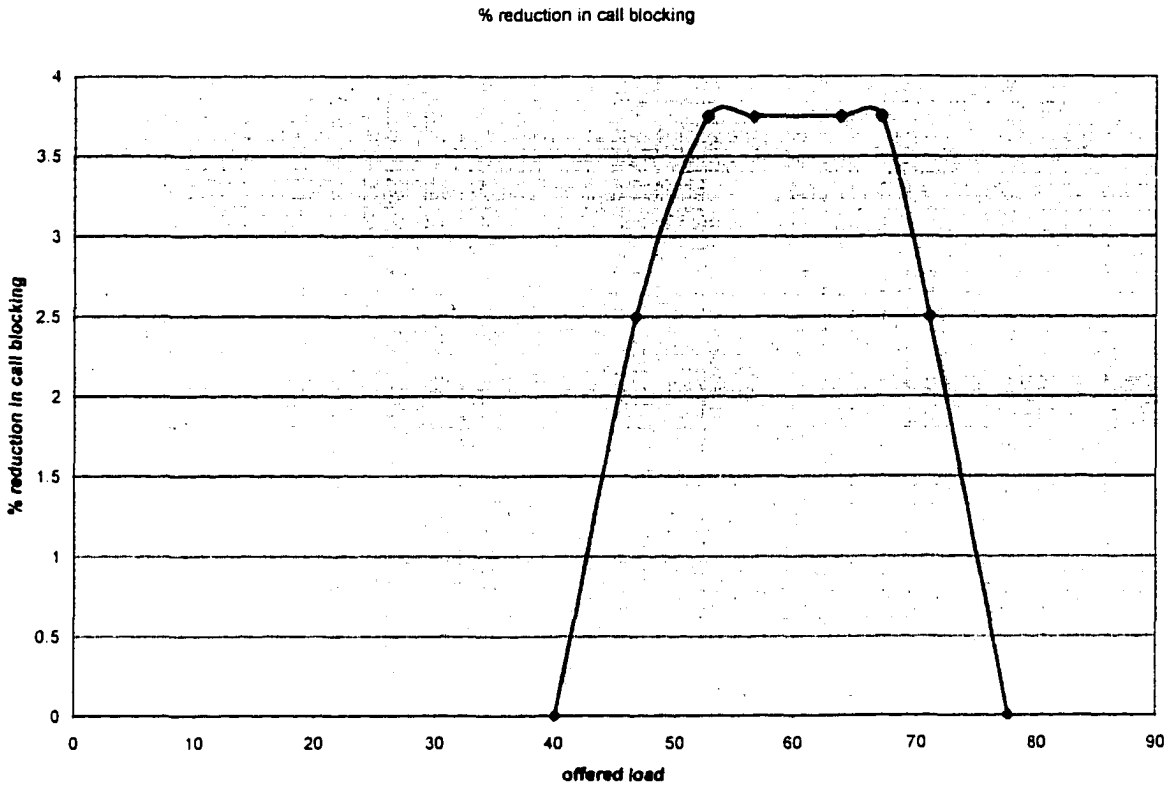


Fig. 4.10 Reduction in Call blocking versus offered load

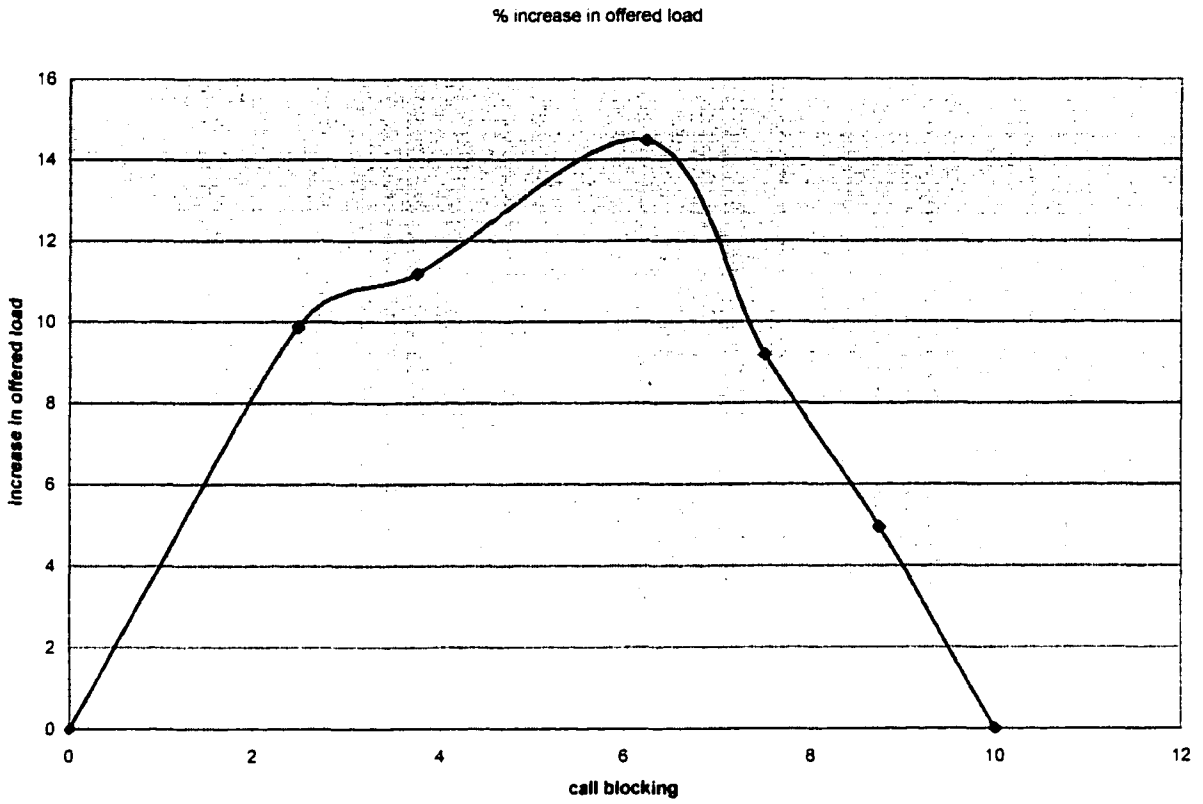


Fig. 4.11 Increase in offered load versus call blocking

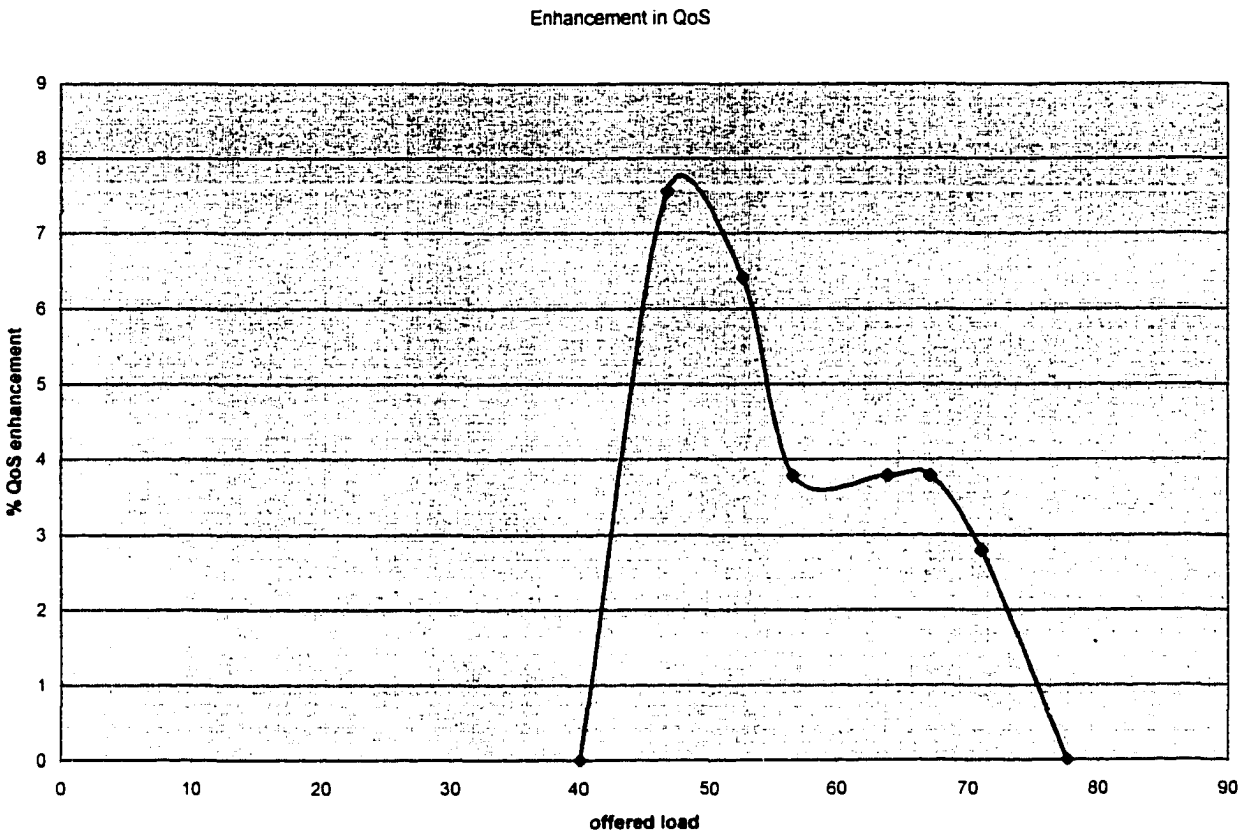


Fig. 4.12 QoS enhancement versus offered load

Above 78%, the acceptor cells can not borrow any cells due to the fact that the algorithm does not allow the borrowing process to take place if the locked channels will cause any current call to be dropped. Therefore, the performance of the channel borrowing algorithm

for an offered load above 78% is exactly the same as without using the algorithm. It is also shown that when using the algorithm and up to an offered load of 53%, the call blocking is kept at 0%.

Fig. 4.10 shows the reduction in call blocking versus the offered load. This figure shows that the advantage of using the suggested algorithm exists when the offered load is in the range from 40% to 78%. The reduction in call blocking reaches its maximum at 53% offered load and is of a value of 3.7% reduction. This reduction is kept flat in the offered load range from 53% to 72%.

Fig. 4.11 shows the increase in offered load for the same call blocking rates. This curve is of significant importance as it shows how much enhancement can be achieved if the system is designed to operate at a certain call blocking rate. It is shown that for a call blocking of 2.7%, there is a 10% enhancement in the offered load. The enhancement keeps increasing till it reaches a peak value of 14.5% at 6.4% blocking rate. Above 6.4% blocking rate, the enhancement starts to decrease till it reaches 0% at 10% blocking rate. Again this is due to the channel blocking affect as the traffic load increases.

Fig. 4.12 shows the enhancement in the QoS offered versus the offered load. This curve shows another contribution of the suggested algorithm. Due to the adaptive QoS platform of the suggested system, the suggested channel borrowing algorithm is also able to enhance the QoS level of existing calls. The enhancement starts when the offered load is at 40%. It reaches a peak value of 7.8% at 45% offered load. It then starts to decrease reaching 0% at 78% offered load.

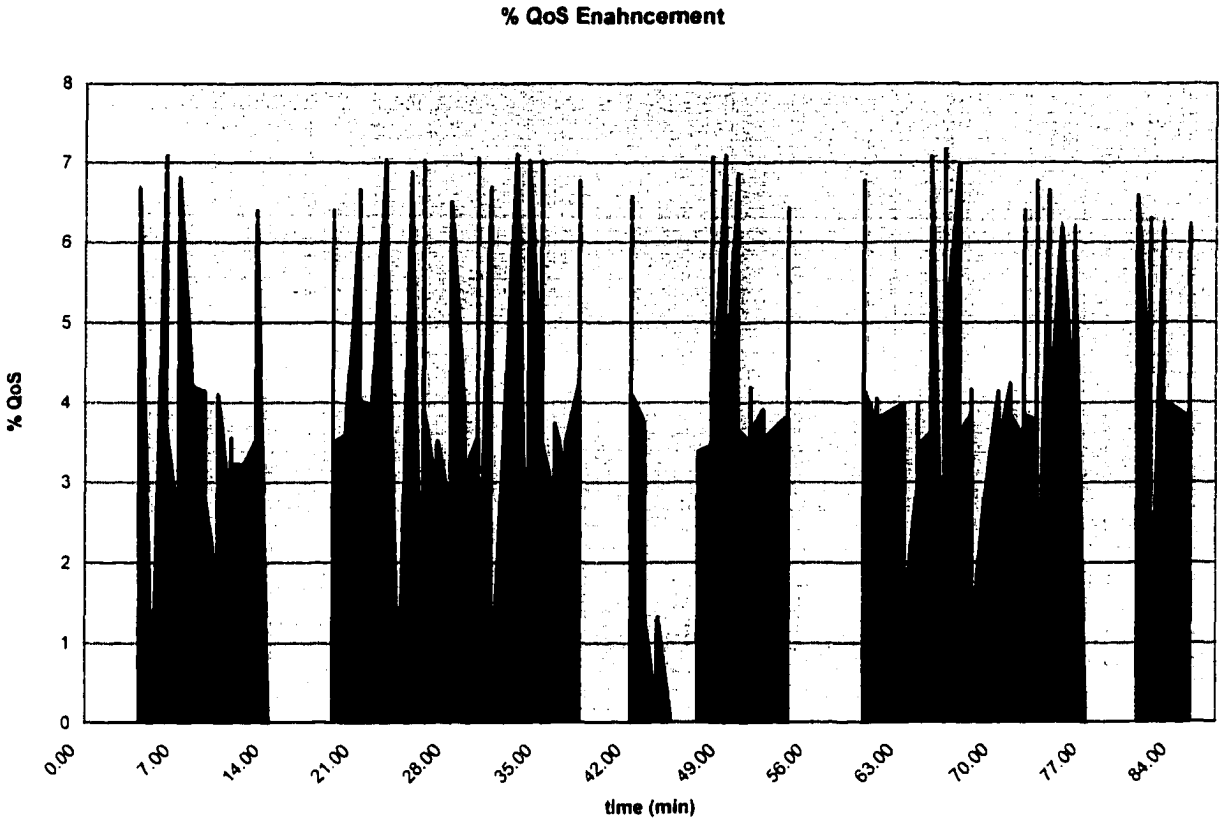


Fig. 4.13 QoS enhancement versus time

Fig. 4.13 shows a sample of the simulator output for 1.5 hours simulation time. The figure shows the enhancement in the percentage QoS level versus time. It is clear that most of the time, the channel borrowing algorithm makes a difference and enhances the QoS level offered to the calls. This is due to the fact that statistically, most of the time the system is operated in the range between 50% to 80% of the offered load and this is when the suggested

algorithm is of great use and advantage. The percentage enhancement varies between 7% and 1%. The average percentage of QoS enhancement is 3.82 %.

Chapter 5: Conclusions and Discussions

In this thesis, we presented an adaptive allocation of resources algorithm for wireless ATM networks. In particular, we used genetic algorithms to maximize the utilization of the scarce wireless bandwidth, while minimizing the call blocking probability. Users are required to define a range of acceptable qualities instead of just a single fixed one. Accordingly, the proposed genetic algorithm was used to adaptively change the bandwidth allocations to the calls with the objective of admitting new calls. This is done at the expense of "temporarily" degrading the quality perceived by the existing calls. However, all calls are guaranteed the minimum quality defined in their perspective profiles. Two separate modules of genetic algorithms were used. Module I was used to do a "fair" bandwidth allocation to each call; whereas Module II re-allocates any available bandwidth that has not been utilized by module I. Performance evaluation results shows significant gains in terms of minimizing the call blocking probability while delivering acceptable QoS levels to the users. Furthermore, the adaptive nature of the algorithm allows it to increase the QoS levels of existing calls once one or more calls depart from the system. Thus adaptability to new traffic situations has been achieved. The application of the proposed algorithm is very flexible making it easy to be applied on a variety of networks.

The thesis also proposes a QoS based channel borrowing algorithm (QCBA) that allows a cell (in a cellular system) to borrow some free channels from any neighboring cell to accommodate new calls. The QCBA is another enhancement layer to the overall suggested call admission control algorithm. The QCBA is specially designed to take advantage of the

adaptive QoS provisioning framework. Simulation study of the QCBA showed significant decrease in the call blocking probability and thus a significant increase in the number of admitted calls also existed. The QCBA also takes channel locking into consideration.

References

[1] Int. Telecommun. Union, ITU-T Recommendation I.371, "Traffic Control and Congestion Control in B-ISDN," March, 1993.

[2] IEEE Pers. Comm., vol.4, no.4, Special issue on IMT-2000, Aug. 1997.

[3] E. Ayanoglu, K. Eng, M. Karol, "Wireless ATM: Limits, Challenges, and Proposals," IEEE Pers. Comm., pp. 18-34, Aug. 1996.

[4] M. Naghshineh, M. Schwartz, A. Acampora, "Issues in Wireless Access Broadband Networks," Winlab '95 Proceedings, 1995.

[5] M. Naghshineh, M. Willebeek-LeMair, "End-to-End QoS Provisioning in Multimedia Wireless/Mobile Networks Using an Adaptive Framework," IEEE Comm. Mag., pp. 72-81, Nov. 1997.

[6] A. Acampora, "Wireless ATM: A Perspective on Issues and Prospects," IEEE Pers. Comm., pp. 8-17, Aug. 1996.

[7] C. Aurrecoechea, A. Campbell, and L. Hauw, "A Survey of QoS Architectures," Multimedia Sys. J., Special issue on QoS Architecture, 1996.

[8] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks," Proc. IEEE, vol. 83, no. 10, Oct. 1995.

[9] L. Wolf, L. Delgrossi, R. Steinmetz, S. Schaller, H Wittig, "Issue of Reserving Resources in Advance," Proc. 5th Intl. Workshop on Network and Operating System Support for Digital Audio and Video, Durham, New Hamshire, April 18-21, pp.27-37, 1995.

[10] C. Aurrecoechea, A. Campbell, A. Eleftheriadis, "Meeting QoS Challenges for Scalable Video Flows in Multimedia Networking," Proc. of the International Workshop on

Network and Operating System Support for Digital Audio and Video, NOSSDAV'95, Durham, NH, April 18-22, 1995.

[11] A. Balachandran, A. Campbell, M. Kounavis, "Active Filters: Delivering Scaled Media to Mobile Devices," Proc. Seventh International Workshop on Network and Operating System Support for Digital Audio and Video, St Louis, May, 1997.

[12] P. Moghe, A. Kalavade, "Terminal QoS of Adaptive Applications," Bell Labs Technical Journal, pp.76-92, April-June 1998.

[13] K. Nahrstedt, H. Chu, S. Narayan, "QoS-Aware Resource Management for Distributed Multimedia Applications," Journal of High Speed Networks, Vol. 7, 3-4, 1998.

[14] K. Nahrstedt, J. Smith, "The QoS Broker," IEEE Multimedia Magazine, Vol. 2, Spring 1995.

[15] F. Garcia, D. Hutchison, A. Mauthe, N. Yeadon, "QoS support for distributed multimedia communications," Proceedings of IFIP/ IEEE International Conference on Distributed Platforms, Dresden, Germany, Feb 1996.

[16] N. Yeadon, F. Garcia, D. Hutchison, D. Shepherd, "Filters: QoS Support Mechanisms for Multipeer Communications," IEEE JSAC, special issue on Distributed Multimedia Systems and Technology, 3rd Quarter 1996.

[17] N. Yeadon, F. Garcia, D. Hutchison, D. Shepherd, "Continuous media filters for heterogeneous internetworking," Proc. of SPIE - MMCN'96, San Jose, CA, Jan 29th-31st, 1996.

[18] N. Yeadon, F. Garcia, A. Campbell, D. Hutchison, "QoS Adaptation and Flow Filtering in ATM Networks," 2nd International Workshop on Advanced Teleservices and High-

Speed Communication Architecture (IWACA '94), Heidelberg, Germany, 26-28 Sept., 1994.

[19] G. Bochmann, A. Hafid, "Some Principles for Quality of Service Management," Distributed System Engineering, Vol.4, No.1, 16-27, 1997.

[20] I. Habib, "Applications of Neurocomputing in Traffic Management of ATM Networks," Proceedings of the IEEE, Vol.84, No.10, pp.1430-1441, Oct. 1996.

[21] Y. Liu, C. Douligeris, "Rate Regulation with Feedback Controller in ATM Networks--A Neural Network Approach," IEEE JSAC, Vol.15, pp.200-208, Feb. 1997.

[22] H. Fahmy, G. Develekos, C. Douligeris, "Application of Neural Networks and Machine Learning in Network Design," IEEE JSAC, Vol.15, pp.226-237, Feb. 1997.

[23] C. Douligeris, G. Develekos, "Neuro Fuzzy Control in ATM Networks," IEEE Comm. Mag., May, 1997.

[24] V. Catania, G. Ficili, S. Palazzo, D. Panno, "A Comparative Analysis of Fuzzy versus Conventional Policing Mechanisms for ATM Networks," IEEE/ACM Trans. On Net., June, 1996.

[25] S. Youssef, I. Habib, T. Saadawi, "A Neurocomputing Controller for Bandwidth Allocation in ATM Networks," IEEE JSAC, Vol.15, pp.191-199, Feb. 1997.

[26] L. Chou, J. Wu, "Buffer Management Using Genetic Algorithms and Neural Networks," Proceedings of IEEE Globecom '95, Singapore, pp.1333-1337, Nov. 1995.

[27] D. Goldberg, "Genetic algorithm in search, optimization and machine learning", Addison Wesley, MA, 1989.

[28] J. Jang, C. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing", Prentice Hall, NJ, 1997.

[29] L. Chou, J. Wu, "Bandwidth allocation of virtual paths using neural-network-based genetic algorithm," IEEE Pers. Comm., vol.1, 1998, p33-39.

[30] P. Ross, D. Corne, "Applications of Genetic Algorithms", AISB Quarterly No. 89, Special Theme on Evolutionary Computing, , pages 23—30, Autumn 1994.

[31] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs," AI Series, Springer-Verlag, New York, 1994.

[32] C. Houck, J. Joines, M. Kay, "A Genetic Algorithm for Function Optimization," NCSU-IE TR 95-09, 1995.

[33] M. Garret, A. Fernandez, "Variable Bit Rate Video Bandwidth Trace Using MPEG Code," Bellcore, ftp://ftp.bellcore.com/pub/vbr.video.trace/, 1992.

[34] M. Garret, "Contributions Towards Real-Time Services on Packet Networks," Ph.D. Dissertation, Columbia University, May 1993.

[35] S. Farouque, "Cellular Mobile Systems Engineering," Artech House, Boston, 1996.

[36] J. Holtzman, D. Goodman, "Wireless Communications, Future Directions," Kluwer Academic Publishers, 1993.

[37] F. Rappaport, "Wireless Personal Communications," Kluwer Academic Publishers, 1993.

[38] I. Katzela, M. Naghshineh, "Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A Comprehensive Survey," IEEE Personal Comm., June 1996.

[39] L. Anderson, "A Simulation Study of Sonle Dynamic Channel Assignment Algorithms in High Capacity Mobile Telecommunications System," IEEE Trans. On Vehicular Tech., Vol VT-22, 1973, p.210.

[40] J. Engel, M. Peritsky, "Statistically Optimum Dynamic Server Assignment in Systems with Interfering Servers," IEEE Trans. On Vehicular Tech., Vol VT-22, 1973, p.203.

[41] K. Chang, J. Kim, C. Yim, S. Kim, "An efficient Borrowing Channel Assignment Scheme for Cellular Mobile Systems," IEEE Trans. On Vehicular Tech., Vol 47, pp.602-608, May 1998.

Publications

1. M. R. Sherif, I.W. Habib, M. Naghshineh, P. Kermani, "An Adaptive Call Admission Control and QoS Management Scheme for Wireless ATM Using Genetic Algorithms," **IEEE wmATM '99**, San Francisco, June 2nd ~ 5th 1999.
2. M. R. Sherif, I.W. Habib, M. Naghshineh, P. Kermani, "An Adaptive QoS Representation and Resource Allocation Scheme for Multimedia and Wireless ATM Networks Using Genetic Algorithms," **IEEE MMT'99**, Venice, Italy, Oct. 6th ~ 8th 1999.
3. M. R. Sherif, I.W. Habib, M. Naghshineh, P. Kermani, "A Generic Bandwidth Allocation Scheme for Multimedia Substreams in Adaptive Networks Using Genetic Algorithms," **IEEE WCNC '99**, New Orleans, Sept. 21st ~ 24th 1999.
4. M. R. Sherif, I.W. Habib, M. Naghshineh, P. Kermani, "An Adaptive Resource Allocation and Call Admission Control Scheme for Wireless ATM Using Genetic Algorithms," **IEEE Globecom '99**, Rio De Janeiro, Brazil, Dec. 2nd ~ 5th 1999.
5. M. R. Sherif, I.W. Habib, M. Naghshineh, P. Kermani, "Adaptive Allocation of Resources and Call Admission Control for Wireless ATM Using Genetic Algorithms," **IEEE Journal on Selected Areas in Communications (JSAC)** special issue on neurocomputing application on wireless communications, February, 2000.
6. M. R. Sherif, I.W. Habib, M. Naghshineh, P. Kermani, "Adaptive QoS Platform in Multimedia Networks", to appear in the **IEEE networking 2000**, Paris, France, May 14th ~ 19th 2000.
7. M. R. Sherif, I.W. Habib, M. Naghshineh, P. Kermani, "A QoS based Channel Borrowing Algorithm", submitted to the **IEEE WCNC 2000**, Chicago, September 2000.

8. M. R. Sherif, I.W. Habib, M. Naghshineh, P. Kermani, "A Lossless Channel Borrowing Algorithm in Adaptive Multimedia Networks", submitted to the **IEEE Globecom 2000**, San Francisco, September 2000.