

Student Work

8-1-2004

A New QoS Renegotiation Mechanism for Multimedia Applications.

Abdelnasser Abdelaal

Follow this and additional works at: <https://digitalcommons.unomaha.edu/studentwork>

Recommended Citation

Abdelaal, Abdelnasser, "A New QoS Renegotiation Mechanism for Multimedia Applications." (2004).
Student Work. 3574.

<https://digitalcommons.unomaha.edu/studentwork/3574>

This Thesis is brought to you for free and open access by DigitalCommons@UNO. It has been accepted for inclusion in Student Work by an authorized administrator of DigitalCommons@UNO. For more information, please contact unodigitalcommons@unomaha.edu.



A New QoS Renegotiation Mechanism for Multimedia Applications

A Thesis

Presented to the

Department of Computer Science

and the

Faculty of the Graduate College

University of Nebraska

In Partial Fulfillment

of the Requirements for the Degree

Masters of Science

University of Nebraska at Omaha

By

Abdelnasser Abdelaal

August, 2004

UMI Number: EP74772

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

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



UMI EP74772

Published by ProQuest LLC (2015). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code



ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

THESIS ACCEPTANCE

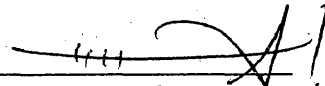
Acceptance for the faculty of the Graduate College,
University of Nebraska, in partial fulfillment of the
requirements for the degree of Master of Science,
University of Nebraska at Omaha.

Committee

Name

Signature

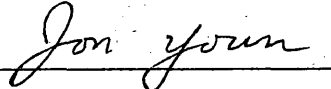
Hesham Ali



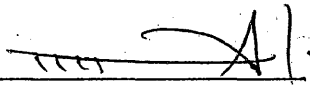
Hamid Sharif



JONG-HOON YOUN



Chairperson (signature)



Date 7/29/14

Co-Chairperson (signature)

Date

(if applicable)

Acknowledgment

My glory, prayers and thanks to Allah who gave me the will and guided me to finish this thesis.

Special note of thanks to my advisor Dr. Hesham Ali for his support, help, and motivation to complete this thesis. I would also express my thanks and gratitude to Dr Hamid Sharif and Dr Jong-hoon Youn for their assistances and feedback.

My great respect, gratitude, and thanks to Dr Hesham El-Rewini for his support and inspiration.

Special thanks to my best friend Sarfraz Hussain for his help and his support.

My thanks and gratitude to my parents, my brothers, and sisters for their prayers and support.

Special thanks, gratitude, and love to my lovely fiancé, Marwah, for her unconditional love, prayers, patience, and support which gave me the inspiration to work hard to finish this thesis.

Thanks to the graduate office and my advisor Carla Frakes for their, smiles, support, and directions.

A New QoS Renegotiation Mechanism for Multimedia Applications

**Abdelnasser Abdelaal, MS
University of Nebraska, 2004**

Advisor: Dr. Hesham Ali

Abstract

While there are a lot of advances in the area of QoS deployment and management over IP networks, there is still a need for a robust QoS renegotiation framework for multimedia applications. A QoS renegotiation framework has three concurrent modules which should be integrated with the least amount of overhead. These modules include a feedback mechanism, a load control mechanism, and a service-response mechanism. This thesis proposes a new feedback mechanism which is based on call rejection notification for QoS renegotiation. The difference between the proposed mechanism and previous approaches is that it uses flow information (not packet information) as a feedback mechanism. The new feedback mechanism provides a better QoS, improves the system performance, and maximizes the service revenue.

Contents

Chapter 1: Introduction.....	1
1.1 QoS and types of applications.....	1
1.1.1 Soft real-time applications.....	1
1.1.2 Hard real-time applications.....	1
1.1.3 Premium Services.....	2
1.2 Resource allocation and classes of service.....	2
1.2.1 Constant Bit Rate (CBR).....	3
1.2.2 Available Bit Rate (ABR) and Unspecified Bit Rate (UBR).....	3
1.2.3 Real Time Variable Bit Rate (RT-VBR).....	3
1.2.4 Non-Real-Time Variable Bit Rate (NRT-VBR).....	4
Chapter 2: Motivation and objective.....	5
2.1 Goals.....	5
Chapter 3: A theoretical background.....	7
3.1 Definitions:.....	7
3.2 Conceptual overview.....	8
3.3 QoS classes.....	9
3.3.1 Absolute QoS guarantee.....	9
3.3.2 Best-effort QoS.....	10
3.3.3 Adaptive QoS.....	10
3.4 QoS parameters.....	11
3.5 Approaches to guarantee QoS.....	15
3.5.1 Service differentiation based schemes.....	15
3.5.2 Resource-reservation-based schemes.....	15
3.5.3 Adaptive QoS.....	17
3.6 Examples for adaptation-based schemes.....	18
3.6 QoS plans.....	22
3.6.1 QoS control plan components.....	22
Chapter 4: DiffServ QoS architecture.....	25
4.1 DiffServ characteristics.....	25
4.2 DiffServ QoS architecture's components.....	29
4.2.1 QoS policing.....	30
4.2.2 Packet classification.....	30
4.2.3 Packet marking.....	31
4.2.4 QoS-aware scheduling.....	32
4.2.5 Traffic shaping.....	33

4.3 Bandwidth broker-based QoS	33
4.3.1 Bandwidth broker's features:	34
Chapter 5: IntServ QoS architecture	36
5.1 IntServ features:	36
5.2 Components of IntServ	38
5.3 Call admission control (CAC)	40
5.3.1 Different approaches to deal with the CAC problem.....	41
5.4 ATM based QoS	49
5.5 DiffServ vs. IntServ	51
Chapter 6: Previous work on adaptive QoS frameworks.....	54
6.1 Previous work	57
6.1.2 Feedback Mechanisms	57
6.1.2 Load Control Schemes.....	61
6.1.3 Service response mechanisms.....	62
6.2 Examples of adaptive QoS aware architectures.....	63
Chapter 7: The proposed approach	65
7.1 The algorithm description and implementation details.....	71
7.2 Call-rejection notification based feedback algorithm.....	72
7.3 The advantages.....	75
Chapter 8: Experimental results and simulation analysis	76
8.1 experiment 1: improving QoS.....	77
8.1.1 Utilization target based IntServ QoS	78
8.1.2 IntServ QoS based on the probability that the needed resources exceed the available resources.....	79
8.1.3. The Incoming flow bandwidth based IntServ QoS.....	81
8.1.4 QoS renegotiation based on call rejection notification feedback.....	83
8.1.5 Comparison study	84
8.2 Experiment 2: Call success versus call rejection ratio.....	85
8.2.1 Link utilization-target-based IntServ	86
8.2.2 The probability that the needed resourced exceeds the available resources....	87
8.2.3 Call admission and rejection for the proposed approach.....	89
8.3 Experiments 3: Call set up and per-flow processing time	92
8.3.1 Utilization target-based IntServ	92
8.3.2 The probability that the needed resources will exceed the available resources	94
8.3.3 Per-flow processing time for the proposed approach	95
8.3.4 Comparison.....	97
Chapter 9: Conclusion and future work.....	100
References:.....	102

List of Figures

<i>Figure 1 : QoS classes</i>	9
<i>Figure 2: An end-to-end resource reservation model</i>	16
<i>Figure 3: DiffServ and BB to concatenate two DiffServ QoS clouds</i>	26
<i>Figure 4: DiffServ as a per-hop behavior</i>	28
<i>Figure 5: DiffServ Packet plan handling</i>	29
<i>Figure 6: Bandwidth Broker based DiffServ QoS architecture</i>	33
<i>Figure 7: An IntServ model</i>	38
<i>Figure 8: IntServ and DiffServ components</i>	52
<i>Figure 9: the components of the QoS renegotiation architecture</i>	56
<i>Figure 10: the link utilization trend after running simulation</i>	69
<i>Figure 11: an adaptive link traffics with adaptive gains</i>	70
<i>Figure 12: An adaptive QoS scheme based on resource reservation and QoS renegotiation</i>	72
<i>Figure 13: Utilization target-based IntServ QoS performance</i>	79
<i>Figure 14: IntServ QoS based on the probability that t the traffics need more resources that what is available</i>	81
<i>Figure 15 : QoS parameters for incoming-flow-bandwidth based IntServ</i>	82
<i>Figure 16: Link utilization with respect to time for the proposed approach</i>	84
<i>Figure 17: comparison among average CLR for the different IntServ approaches and the adaptive approach</i>	85
<i>Figure 18 : call admission and rejection for Utilization target based IntServ</i>	86
<i>Figure 19 : call rejection and admission with respect to the probability that the needed resource will exceed the available resources</i>	89
<i>Figure 20 : Number of admitted and rejected calls for the proposed approach</i>	91

List of Tables

<i>Table 1: QoS architecture and the proper class of service and application.....</i>	<i>4</i>
<i>Table 2 : technology-based QoS parameters.....</i>	<i>14</i>
<i>Table 3: Comparison between IntServ and DiffServ QoS architecture (* indicates area of research)</i>	<i>53</i>
<i>Table 4: utilization target-based QoS.....</i>	<i>78</i>
<i>Table 5: IntServ QoS based on the probability that the needed resources exceeds the available resources</i>	<i>80</i>
<i>Table 6: The incoming flow bandwidth based IntServ QoS.....</i>	<i>82</i>
<i>Table 7: QoS parameters for the proposed approach.....</i>	<i>83</i>
<i>Table 8: QoS parameters for IntServ different approaches comparing with the proposed approach.....</i>	<i>84</i>
<i>Table 9 : call admission and rejection for Utilization target based IntServ.....</i>	<i>87</i>
<i>Table 10 : call rejection and admission with respect to the probability that the needed resource will exceed the available resources</i>	<i>88</i>
<i>Table 11 : Number of admitted and rejected calls for the proposed approach.....</i>	<i>90</i>
<i>Table 12 : Flow processing time with respect to link utilization</i>	<i>92</i>
<i>Table 13 : Per-flow processing time with respect to PNREAR.....</i>	<i>94</i>
<i>Table 14 : per-flow processing time for the proposed approach with respect to time intervals.....</i>	<i>96</i>

Acronyms:

CAC Call admission control

CLR Cell loss rate

DiffServ Differentiated Services

IntServ Integrated Services

PBACA Parameters based call admission control algorithms

MAs multimedia applications

MBACA measurements based admission control algorithms

QoS quality of service

WNs wireless networks

Introduction

Internet subscribers increase 30% per month in USA. In addition, statistics show that only 39% of American population use cell phones while about 60% of Europeans and Japanese have cell phones. Therefore, there is a great potential growth in the American communication industry. This potential growth in wireless communication industry will not only include increasing the number of users, but also includes developing new applications, improving current applications, and improving the quality of service. Examples of new applications are movie trailers and entertainment messages. In addition, wireless network services should be scalable and should be able to reach remote areas. The scalability concept is so important so that it encompasses the most challenging topics in wireless networking: extendibility and quality of service. Scalability means increasing the performance of the network linearly with the increasing number of nodes. This thesis proposes a new feedback mechanism for QoS renegotiation for multimedia applications. This feedback mechanism is based on call rejection notification.

Thesis organization:

This thesis has been divided into 9 chapters. Chapter one is an introduction which includes types of applications, classes of services and their QoS requirements. Chapter two discusses the motivations and goals. Chapter three is a theoretical overview. It includes some important definitions, and concepts, QoS classes, QoS parameters, different approaches to guarantee QoS, and QoS plans. Chapter four presents one of the most widely known approaches for guaranteeing QoS; DiffServ QoS architecture. It presents its characteristics, its components, and bandwidth broker. Chapter five introduces IntServ QoS architecture. It discussed IntServ features, components, and call admission control. In addition, it Discusses ATM as an off-line QoS negotiation technology and a comparison between DiffServ and IntServ. Chapter six introduces the new adaptive QoS approach. It discusses previous work on feedback mechanisms, previous work on load control schemes, and previous work on service response schemes. Chapter 7 introduces the proposed approach. Chapter 8 is experimental results. The first experiment is to test the QoS improvement for the new approach. The second experiment is to examine call rejection and admission percentage. The third experiment is to examine the call processing delay. Chapter 9 is conclusion and future work.

Chapter 1: Introduction

1.1 QoS and types of applications

1.1.1 Soft real-time applications

The soft real-time applications can be delivered in a best-effort delivery framework. In other words, such services do not require QoS guarantees (unless specified). An example of soft real-time application is email application. Soft real-time services are more scalable than the hard real-time services. DiffServ is a suitable framework of soft real-time applications.

1.1.2 Hard real-time applications

Hard real-time applications need guaranteed QoS. This property comes from the application characteristics or is requested by the service contractor. Examples of hard real-time applications are audio and video conference applications. Audio applications can tolerate higher cell loss rate and limited bandwidth. However, in order to tolerate limited bandwidth, multimedia applications require adequate information about the available resources through an adaptive QoS framework before the call set-up. Providing this information enables the application to be adapted with the available resources. On the other hand, video conference applications have more options than audio applications. Specifically, beside its ability to adapt to the available bandwidth, video applications could be multi-casted. Multi-casting is another approach to deal with limited resources. Although audio and video applications could adapt to available resources, they are too

sensitive to jitter. Therefore, minimizing the jitter rate is crucial for such applications. IntServ is a suitable QoS architecture for hard real-time applications. However, the current standards of IntServ do not support multicasting.

1.1.3 Premium Services

Premium services require peak rate allocation and high-priority queue in routers. Therefore DiffServ is suitable for this kind of service.

1.2 Resource allocation and classes of service

Resource reservation is very important for a hard QoS guarantee. There are two methods of resource allocation: peak-rate-based allocation, which uses a single QoS parameter for resource allocation, and probabilistic allocation. The probabilistic allocation approach is based on assuming some variation in the expected cell-loss rate and delay rate. The real-time services are called deterministic guaranteed services. This class of service requires a very small packet-loss rate and a hard delay bound. Call admission control algorithms (CAC) are two categories: peak bandwidth allocation and statistical allocation. The peak bandwidth allocation or the non-statistical allocation is used if the required QoS (usually delay rate) is deterministically bounded. The statistic resource allocation is used to deterministically or statistically guarantee the QoS. The CAC decision is very easy in case of the peak rate-based approach. However, this approach does not maximize the utilization of the available resources. In statistical resource allocation approach, resources are not allocated based on the peak rate. Specifically, statistical resource allocation

considers traffic burstness so that the allocated resources are less than the peak rate. Therefore, the sum of all peak rates may exceed the available resources.

1.2.1 Constant Bit Rate (CBR)

Real-time continuous streams, which does not tolerate high cell-loss rate, requires CBR. This class of service needs resource reservation. CBR real-time applications require connection-oriented networks. Therefore, IntServ is the most suitable QoS architecture for this class of service. Examples of suitable applications for this class are interactive video and audio conferences and circuit emulation applications.

1.2.2 Available Bit Rate (ABR) and Unspecified Bit Rate (UBR)

UBR class of service is suitable for best-effort and non-real time applications. It does not require a QoS guarantee. Usually, these applications require a lower cell-loss rate. Examples of this class of service are data applications. DiffServ is the most suitable QoS architecture for this class of service. In addition, an adaptive QoS framework with a controlled feedback mechanism could fit this class of service as well

1.2.3 Real Time Variable Bit Rate (RT-VBR)

This class of service needs a reserved bandwidth. Therefore, adaptive QoS with a feedback mechanism in a controlled environment is suitable for this class of service. It has peak rate, sustainable rate, and maximum burst size. Therefore, VBR is suitable for compressed audio and video applications and any applications other than video conferences. Audio and video applications must maintain the short-term loss rate and

keep a constant long-term loss-rate target. VBR needs a connection-oriented service. Controlled services are for adaptive/tolerable applications (applications with cell-loss and delay requirements). Therefore, IntServ QoS is the most suitable architecture.

1.2.4 Non-Real-Time Variable Bit Rate (NRT-VBR)

This class is similar to RT-VBR and it can tolerate a higher delay rate. The most suitable applications are video playback and transaction processing. Adaptive QoS in a controlled environment is suitable for this kind of application.

QoS architecture	Class of service	Applications
IntServ	CBR and ABR	Interactive video and audio conferences and circuit emulation, and premium applications
DiffServ	ABR	Data and non real time applications
Adaptive QoS	NRT-VBR and RT-VBR	Compressed video and audio applications, video playback, data applications

Table 1: QoS architecture and the proper class of service and application

Chapter 2: Motivation and objective

Statistics shows that fewer than 40% of Americans use cell phones. However, 60% of the population uses cell phones in Europe and Japan. Wireless applications are increasing rapidly, e.g., movie trailers and entertainment messages. However, WNs are a resource-poor environment with a lot of overhead and high error rate. These characteristics need an end-to-end QoS guarantee using resource reservation scheme and control scheme. This architecture should be supported by a feedback mechanism to provide adaptive QoS. Adaptation is essential to achieve fairness, efficient use of resources, a better QoS guarantee, and QoS scalability. Obviously, in a mobile environment, resource reservation and end-to-end load control are essential for guaranteed QoS. Multimedia applications (MAs) are increasing rapidly. These types of services require guaranteed QoS. Fortunately, audio and video applications can renegotiate their required QoS. Specifically, they can tolerate higher delay and higher cell-loss rate. In addition, audio and video applications can adapt to the available bandwidth once applications have been informed with the available bandwidth at the connection set-up.

2.1 Goals

This work aims at developing an adaptive QoS architecture based on a feedback mechanism and a periodical resource estimation scheme. The feedback mechanism depends on call rejection notification to inform MAs of the available resources and the network conditions. This feedback mechanism will help MAs to renegotiate their

required QoS. This feedback mechanism works in an end-to-end controlled environment. Basically, in high congestion levels, the MAs are required to renegotiate their QoS to be supported with the minimum acceptable QoS. In low-congestion-level states, the system will provide MAs with the minimum tolerable QoS parameters. In the normal conditions, the network provides MAs with their required QoS.

Chapter 3: A theoretical background

This chapter presents a brief theoretical foundation of current QoS architectures.

It discusses the conceptual framework for the QoS model in terms of QoS dimensions, different approaches to guarantee QoS, examples for the current QoS architectures, and an example for a QoS-aware transmission technology. In addition, it emphasizes both the differences between the traditional QoS model and the adaptive QoS model.

3.1 Definitions:

Quality of Service (QoS): For the purpose of this work, Quality of service refers to the characteristics of service delivery provided to the packet such as bandwidth, delay, loss rate, and jitter.

Flow: In the context of this thesis, a flow is defined as a set of packet streams a network node receives and all these flow streams have the same flow address and all belong to the same admission request. Specifically, it means multimedia (audio and video) flows.

Heterogeneous networks: in the context of this thesis, heterogeneous networks mean multiple network clouds which include wired and wireless domains, different applications, and bandwidth and delay variation.

3.2 Conceptual overview

Recently, QoS has become a topic of a lot of research. QoS has been defined as the degree of satisfaction the user gets from the service performance [3], as well as, the traffic-performance matrix which is to be achieved. This QoS matrix must be routed from end to end. The QoS routing is finding an end-to-end path to support such traffic matrix. This routing protocol should be fast and is loop-free. In addition, it has to have low overhead.

An adaptive Quality of Service model is needed to adapt to network conditions and application characteristics. A comprehensive survey of different QoS architectures for WNs has been introduced in [1 and 2]. In addition, a benchmarking WNs QoS has been introduced in [3].

QoS becomes an issue only when the network is congested or overloaded. Congestion arises because of either insufficient network resources or improper resource allocation (some resources might be over utilized and others might be underutilized). With QoS renegotiation, which might lead to QoS rerouting, the imbalance of resource allocations problem could be solved.

A QoS model should differentiate between traffics based on the degree of reliability, delay tolerance, and the required bandwidth. A QoS model should provide flow rejection and admission notification.

QoS should be configurable, maintainable, measurable, and predictable from end to end. Moreover, QoS parameters must be specified quantitatively, statistically, or qualitatively at the configuration time. For maintenance of the QoS, parameters must be mapped for each packet not only from network cloud to another, but also from end to end. Therefore, the QoS in the network is the QoS of the weakest segment or hub in the network.

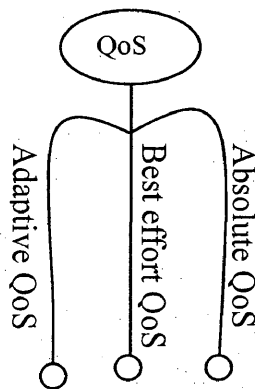


Figure 1 : QoS classes

3.3 QoS classes

QoS could be put into three classes as illustrated in Figure 1.

3.3.1 Absolute QoS guarantee

IntServ QoS architecture is an ideal example of this QoS class. In this class, each flow is given special treatment. This per-flow treatment requires an end-to-end control mechanism and per-flow state maintenance. In addition, absolute QoS guarantee requires resource-reservation mechanism and an admission-control scheme. Its services are

differentiated (usually based on the application's characteristics). Obviously, this class of QoS has a lot of overhead and it is not scalable.

3.3.2 Best-effort QoS

This category represents the traditional IP transmission technology which is based on best-effort delivery. In this approach, all traffics are treated equally either in scheduling or in case of dropping packets when congestion occurs. Of course, best-effort QoS class can be always accepted without processing in the QoS model.

3.3.3 Adaptive QoS

In adaptive QoS, specified QoS parameters could renegotiate and change dynamically to adapt with the application characteristics, the congestion levels in the network, the available resources, and any unforeseen conditions. Usually, this adaptation has an upper-bound and a lower-bound parameters. The flow is admitted only if the lower-bound parameters of the required QoS could be guaranteed in the long run. One of the advantages of adaptive QoS is achieving fairness among different traffics. In other words, it is better for applications to renegotiate their QoS requirements rather than get rejected. Therefore, adaptive QoS is more scalable than absolute QoS. The second advantage is the application's ability to adapt to path characteristics. Resource reservation is not a must for this approach. However, it needs a feedback mechanism to provide routers with information about the network conditions and available resources.

3.4 QoS parameters

The QoS n-tuple (QoS_r, QoS_o) should apply. QoS_r is the QoS required by the application or the user, QoS_o is the offered QoS in the system. In the absolute QoS model, this tuple is hard which means it must be met. But in adaptive QoS specially application based adaptation; by QoS renegotiation the QoS_r could be renegotiated to meet QoS_o.

Equation 1: $QoS_o = f(r, s, m, l)$

In Equation 1, QoS_o is a positive function of R and a negative function of s , m , and l . Where, r is the available resources (usually the effective bandwidth plus the buffering capacity). s is the degree of the network and applications seamless. m is the mobility pattern of the mobile host, and L is the traffic load.

QoS parameters could be classified based on the user's prospective or technology prospective [1] or based on wired and wireless characteristics. These QoS parameters are specified for each application in the QoS matrix at the contract time or at the configuration time. This process is called *QoS specification*.

- **Technology-based QoS parameters:** These parameters include delay, response time, and jitter. Sometimes these three parameters are called timeliness parameters. The second category is resource-based parameters. This category includes the following parameters: the required resources for application, the available resources in the system, and the

transmission rate or effective bandwidth. Of course, the available resources are a main factor especially in a poor-resources environment and a dynamic resource-allocation model. The third category is reliability parameters. This category has three parameters: the mean time to failure (MTTR) which is the operation time between failures, mean time between failures (MTBF) which is the time from failure to restarting the system, and the availability percentage which is $MTBF / (MTBF + MTTR)$. The last QoS parameter is the cell lose rate (See Table 2).

- **User-based parameters:** These parameters could be classified into three categories. The first category is perceived QoS which includes picture details, picture color accuracy, video rate and smoothness, audio quality, and video/audio synchronization. The second category is criticality. This category includes giving priority to services. The third category is cost (either per user or per unit). The next category is security which includes confidentiality and integrity, authentication, and non repudiation of sending or delivery.

Traffic parameters: this category includes peak cell rate (PCR), sustainable cell rate (SCR), maximum burst rate (MBR), minimum cell rate (MCR), and constant bit rate (CBR).

- **End-to-End parameters:** this category includes cell-delay variation (CDV) which is called Jitter sometimes, maximum cell transfer delay, and cell-loss rate (CLR). QoS guarantees a specific end-to-end transmission performance at the ATM layer. The QoS parameters are either negotiated or non-negotiated parameters. The non-negotiated

parameters are cell mis-insertion rate, severely errored cell block rate (SECBR), and cell error rate (CER). Examples of the negotiable QoS parameters are cell-loss rate (CLR), maximum cell transfer delay (maxCTD), peak cell rate (PCR), and peak-to-peak cell-delay variation (PTPCDV). FTP applications are sensitive to packet loss but it can tolerate higher packet delay. On the other hand, Telephone calls are sensitive to delay but it is tolerant to low cell loss.

There are special QoS parameters for Wireless networks. These QoS parameters are channel-error rate, handoff-blocking rate, and new-call-blocking rate.

Category	Parameter	Description/ example
Timeliness	Delay	This delay includes switching delay, access, and queuing delays
	Response time	The time from submitting a request to receiving the reply
	Jitter	Variation in delay
Bandwidth	Channel bandwidth	Sometimes it is called the available resources, which includes channel capacity and buffer size
	Application's required bandwidth	
	Transmission rate	Or the effective bandwidth
Reliability	Cell loss rate	Losing cells because of congestion, fading, or errors
	Mean time to failure(MTTF)	Time between failures
	Mean time to repair(MTTR)	Time from failure to restarting the system
	Availability percentage	$MTTF / (MTTF + MTTR)$

Table 2 : technology-based QoS parameters

3.5 Approaches to guarantee QoS

There are three major categories of schemes to address guaranteed QoS.

3.5.1 Service differentiation based schemes

Most of these schemes are implemented under the notion of DiffServ QoS architecture.

DiffServ is a per-hop behavior. The concern about DiffServ is whether a given packet will be treated the same in all network clouds.

DiffServ offers two kinds of services, best-effort and virtual leased lines. In virtual leased lines, clients need absolute bandwidth allocation and must be treated differently than other users.

DiffServ does not guarantee quality, does not create bandwidth and it is a zero-sum game.

In other words, giving some classes' especial treatments will be at the expense of other classes.

3.5.2 Resource-reservation-based schemes

Most of these schemes are implemented in the notion of IntServ QoS architecture.

IntServ requires end-to-end resource reservations as illustrated in Figure 2. These schemes can use one or all of the following IntServ components; call admission control algorithms (CAC), resource reservation protocols, bandwidth broker, signaling schemes, and monitoring protocols. Figure 2 shows that for resource-reservation-based QoS, end-to-end resource reservations are required before call set-up. Therefore, there is no optimal utilization for the available resources.

Guaranteeing QoS for multimedia traffics is very difficult for the following reasons: it is difficult to predict the traffic characteristics, the traffic may be very burst which needs a very conservative peak rate. In the offline allocation, the call admission decision is made before the connection starts. Off-line resource allocation could be static reservation or dynamic reservation (which allows QoS renegotiation during the connection). Usually, static allocation uses peak rate allocation, which is suitable for real-time applications. The on-line method periodically renegotiates the QoS parameters and re-allocates resources based on the feedback and the predicted behavior of the traffic. Although this method avoids most of the off-line approach problems, its overhead and complexity reflects excess renegotiation and re-allocation.

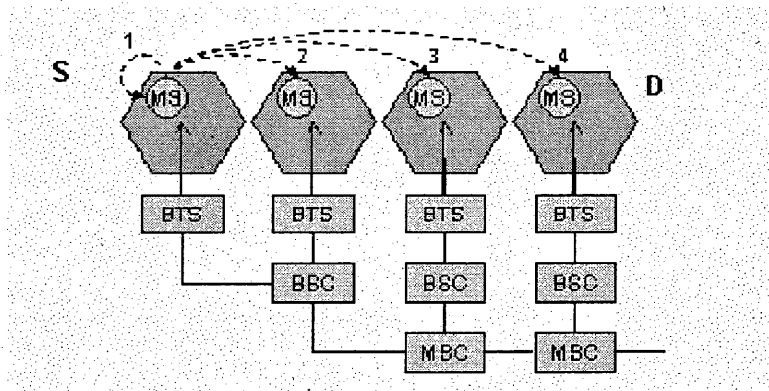


Figure 2: An end-to-end resource reservation model

A dynamic advance resource-reservation scheme is offered in [4]. In this approach, a time-slot manager has been used to guarantee that the committed resources will not exceed a specified limit and to estimate the reserved (and unused) bandwidth. To

guarantee QoS for MAs, some reservation schemes [5] reserve resources in all surrounding cells, leading to inefficiency in resource utilization and a high connection blocking rate.

An adaptive QoS architecture needs a specification for the application characteristics, a controlled feedback mechanism, and QoS renegotiation ability.

3.5.3 Adaptive QoS

Real time applications can adapt to available bandwidth if the application is aware of available bandwidth at the connection establishment. Many approaches [6] have been proposed to support adaptive multimedia over WNs. At the application layer level, real-time application can be encoded to adapt to heterogeneous networks. At the transport layer, resource reservation could be made to support end-to-end QoS. At the network layer, techniques to support mobility and QoS-aware routing have been proposed. At the data-link layer, MAC has been modified to support QoS.

Adaptive QoS (AQoS) tries to guarantee a matrix of QoS parameters over a heterogeneous network and the QoS matrix changes dynamically per call and based on the system conditions. Therefore, absolute QoS is suitable for wired networks, and wireless networks require adaptive QoS.

The benefits from adaptation are: improving QoS, decreasing call blocking rate and handoff dropping probability, achieving fairness among multi-service networks, and increasing utilization of available resources.

3.6 Examples for adaptation-based schemes

These schemes are to provide per-connection QoS based on the available resources, application characteristics, and network conditions. The most adaptive technology is ATM which provides per-flow QoS because ATM can provide bandwidth on demand. However, ATM provides offline adaptation and it does not support online adaptation. There are many approaches for adaptation. In the sender-based adaptation, the sender can adapt the transmission rate to the available resources (which is suitable for audio and video applications). One of the famous methods of sender-based adaptation is video compression adaptation with limited resources. The second approach is the receiver-based adaptation; transmission techniques and encoding mechanisms are the most famous examples of receiver-based adaptation.

Levels of adaptation

There are many kinds and levels of adaptation [7]. There is a sender and receiver-based adaptation. There is application-based adaptation and there is network layers-based adaptation.

Network layer-based adaptation: at this level classifying the services based on their required QoS and reserving resources for the applications flows is made. The network adaptation level is responsible for mapping the QoS metrics. Mapping QoS includes assigning each packet with its required QoS parameters, mapping the QoS parameters through the network path and the network layers from end-to-end (from the user's application layer to the sender's application layer).

An adaptive resource reallocation is used to maximize the utilization of the available resources. Also, at this layer, routing should be adapted to the mobility and handoff. In addition, call admission control algorithms are one of the important network layer-based adaptation methods.

COMET [8] and RDRN [9] provide handoff control mechanisms that adapt to seamless networks.

Application-based Adaptation: there are two kinds of applications: real time and non-real time. Non real time applications (usually data) are not sensitive to delay and bandwidth limitations. Although they are sensitive to high jitter rate, real-time applications could adapt to varying bandwidth and delay. This kind of adaptation ensures that the application adapts to limited resources. Specifically, voice and video can be encoded differently. In other words, encoding parameters can be modified dynamically as a response to a changing environment. Encoders use different compression rates to get different QoS levels. Adaptive encoding approaches could be used to smooth the bandwidth of the encoding streams. Adaptation could be done by encoding [10], scaling [11], or filtering techniques [12].

An adaptive multimedia approach in wireless IP networks has been discussed in [13]. In addition, there are other application-based adaptation techniques such as: video

transcending, multiple video streams, and scalable coding schemes, storage overhead, and reducing processing requirements.

Application-based adaptation scheme is introduced in [41]. In this approach, the flow state interacts with the network QoS. In other words, applications dynamically choose which flow-state they can use based on a feedback mechanism. A dynamic application adaptation scheme has been introduced in [15]. In this scheme the application sends a signal to the reservation scheme to reserve its QoS.

For example, voice applications can be transmitted using bandwidth varying from 8kbps to 128kbps. Moreover, different encoding mechanisms can be used based on the network conditions.

Discrete Cosine Transformation (DCT) compression is another adaptation scheme. In addition, scheduling and priority en-queue can be used for such kinds of adaptation.

In the COMET [8] project, it has an adaptive application layer that guarantees acceptable QoS and use the residual bandwidth to enhance the QoS when resources are released.

The Link layer-based adaptation: most of the adaptation mechanisms over the link layer are done over MAC layer. MAC protocols organize shared resources in a way that achieves fairness among different applications. The MAC layer solutions are trying to minimize the control overhead. In [16] a discussion about the relationship between the MAC protocol and QoS has been introduced. To improve the QoS parameters, a proper

MAC protocol is required to maximize utilization of available resources while providing the QoS. It enables the end-user to connect with the satellite network and use its resources. The traditional FDMA and TDMA are bandwidth wastage. Therefore, it is not proper to use them in a poor bandwidth environment. There are two major MAC protocols for Satellites: Multi-Frequency TDMA and CDMA. MF-TDMA is based on the bandwidth reservation scheme for all kind of services. On the other hand, CDMA does not need pre-allocation for resources. It manages the resource allocation among different users.

Therefore, the application-based adaptation is preferred because it does not generate much overhead and it can support end-to-end QoS. MAC protocol manages the available resources among different users. It decreases implementation complexity, decreases the delay, and increases the effective bandwidth in the channel. Therefore a QoS aware MAC protocol is crucial for guaranteeing QoS. In a per-hop QoS model the mobile hosts communicate with each other either directly or through the base station. On the other side, in a multi hop network, clustering is used for bandwidth management, nodes communications, and feedback. QoS routing is important for mapping the QoS metrics from end to end. The cluster head gets acknowledgement of packets delivery and VC reservation.

3.6 QoS plans

There are two dimensions for any QoS model. The first dimension is the packet plan which includes policing, packet classification, marking, traffic shaping, queuing, and scheduling. Policing function is responsible for dropping packets that do not meet the QoS specification. Shaping function buffers packets that do not meet QoS specification. Marking marks packets with their designated priority. The packet plan will be discussed in details next chapter because it is clear in DiffServ architecture

The control plan includes QoS signaling, resource management and reservation, QoS routing, congestion control, services differentiation, clustering, and mobility management. The resource management category includes call admission control and channel reservation, decreasing control overhead, and buffering. However, all the control modules must be integrated with least overhead. The main goal from the control plan is reserving channels and routing the QoS matrix through a proper path, and increasing the efficiency of bandwidth utilization. Usually the control plan is a flow-oriented function but the packet plan is a packet-oriented function. IntServ explicitly implements the control plan while processing the flows. DiffServ implicitly enforces the control plane while treating packets differently.

3.6.1 QoS control plan components

The control plan is responsible for channel allocation, resource allocation, choosing the proper path, QoS routing, and call setup. The QoS control plan is responsible for managing QoS parameters from end to end.

Channel reservation and path setup:

Channel and bandwidth reservation is crucial for multimedia applications to guarantee QoS. This reservation could be advance reservation or dynamic reservation. Advance reservation is always offline, but dynamic reservation could be online or offline. However, a good resource-reservation algorithm should give a higher priority to handoff handling than to accepting a new connection. Channel reservation has been discussed in [17], [18], and [19]. In [18] the channel is reserved based on the probabilistic analysis of the mobility of the user at all neighboring cells. [17] Provides higher QoS by reserving channels at all neighboring cells simultaneously. However, this approach wastes bandwidth. In [19] reservation is done based on the mobility information, the current position of the user, and the power level that is received by the user from its adjacent cell. This scheme reduces both new blocking rate and handoff blocking rate compared to other reservation schemes. In [20] a movement probabilities based reservation scheme is introduced in multimedia wireless networks.

Multicasting is a good way to increase the number of users or the delivered packets at the end point without increasing the bandwidth. IntServ can support multicasting but the current DiffServ architecture cannot support multicasting. Multicasting is very important to audio and video applications. Therefore any measurements to the effective bandwidth should consider the gains from multicasting [21]. Therefore, multicasting is a good

mechanism to enhance the scalability of the network, especially a wireless ATM networks.

Network clustering and partitioning:

Networks work under the concept of clouds or clusters. Partitioning the network into optimal sizes of clouds or clusters is very important for resource management, QoS shaping, call admission control, and congestion control. Clustering is partitioning the mobile nodes into groups whose size is optimal. The optimal cluster size is a function of the mobility model, available resources, and expected traffic load. However, the optimal clustering goal is optimizing the utilization of the available resources. Clustering has been discussed in [22, 23].

A novel framework for dynamic network clustering for mobile nodes in wireless ad-hoc networks in which the probability of path availability is bounded has been introduced in [22]. WAMIS [24] uses clustering to enhance CDMA code separation. This architecture uses graph-coloring in clustering and code assignment.

Chapter 4: DiffServ QoS architecture

The internet in USA is growing at a rate of 20% per month. Therefore, the QoS should be scalable to meet these growth requirements. Current IP QoS architectures do not provide mechanisms to support complete and proper service response to network conditions and available resources.

In [25], there is a descriptive detailed DiffServ model from ingress node to egress node.

Current DiffServ standards have been discussed in [26].

4.1 DiffServ characteristics

Service aggregation and differentiation based architecture: DiffServ differentiates and classifies packets at network edges into classes and treats them based on their priority and class of service. Traffic classification could be done at higher layers to get some specific degree of granularity. However, higher granularity will lead to higher delay. Of course, dealing with packets as classes reduces the overhead of per-flow control and provides a specific bounded delay. DiffServ aggregates many traffic streams into a small number of service classes. This aggregation process is based on per-hop behavior. In addition, DiffServ does not require signaling other than the specified service class in the DSCP of the packet. That indicates DiffServ simplicity and scalability

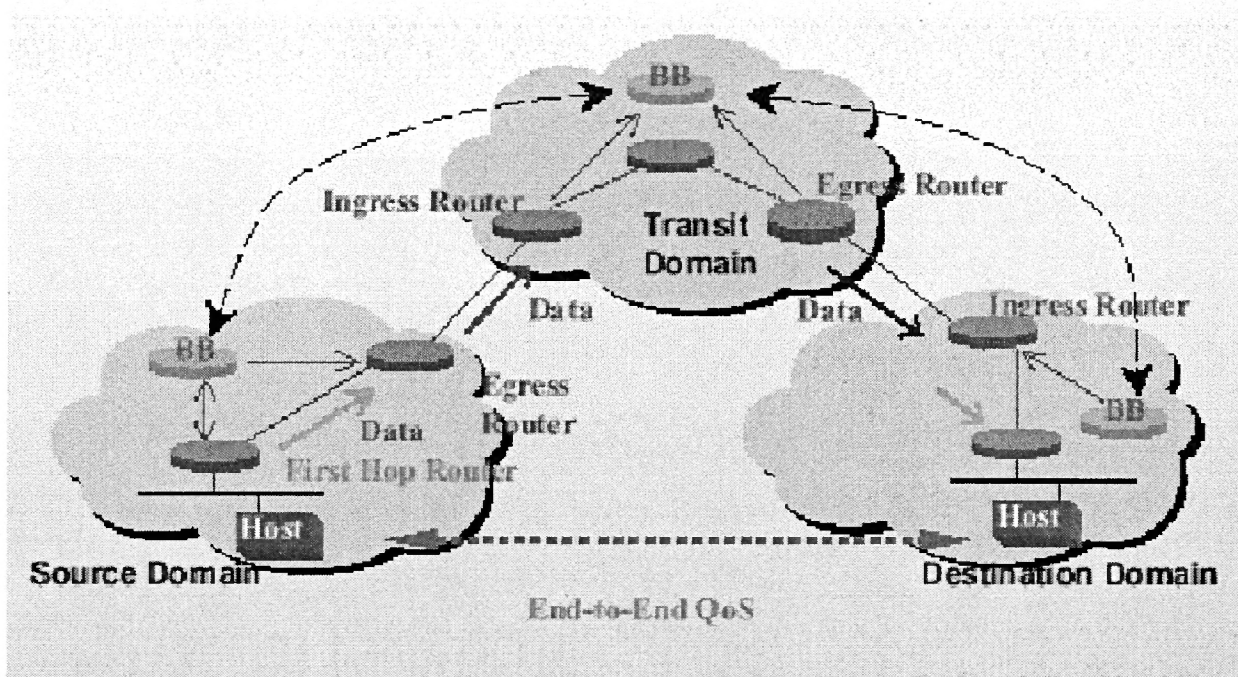


Figure 3: DiffServ and BB to concatenate two DiffServ QoS clouds

DiffServ has been developed as a need for a scalable, simple, and easy deployable QoS for IP networks without end-to-end per-flow signaling. In other words, it aggregates packets into aggregates of flows and treats them based on their specified level of service in the packet header. It provides QoS by classifying packets into classes of services. It prioritizes packets in a way which meets the user's QoS requirements at the configuration time. In other words, it marks packets with codes that indicate its class. The higher-priority packets get forwarded first and the low-priority packets dropped first. It separates packet processing from the routing process.

The DiffServ network administrator determines the level of service class at the configuration time according to the user's specification. In addition, traffic classification

can be done in any of the higher layers to get some specific degree of granularity. However, higher granularity means higher delay. Of course that reduces the overhead of per-flow control and provides a specific bounded delay. It avoids the complexity of maintenance of per-flow state information and it shifts the control process only to the edge routers.

Scalability: DiffServ enforces QoS management (classification, policing, marking, and shaping) on the edge routers [27, and 28]. DiffServ provides more-scalable QoS for IP networks than IntServ because of the following reasons: (i) it decouples the packet processing function from the control function. (ii) Only edge routers manage and process QoS management. DiffServ avoids the scalability problem of IntServ by performing packet classification and marking only at the edge routers. The core routers simply forward packets based on the specified class of service identified in the packet header.

DiffServ QoS marks packets to trigger a proper response from a router. Therefore, the calculation processing and memory consumption do not increase in proportion to the increase of traffic load.

Decoupling the control plan from the data plan: DiffServ decouples the control plan from the data plan by marking each packet with its class of service. This is an implicit implementation for a control scheme.

Stateless QoS architecture: DiffServ QoS is called stateless QoS. In other words, DiffServ core routers do not maintain per-flow state information about current flows. It avoids the complexity of maintenance of per-flow state information. The problem with such a stateless QoS architecture is that it does not guarantee flow admission and it does

not provide flow-rejection notification. It is the application's responsibility to figure out if its admission request has been honored or not.

DiffServ is a per-hop-behavior: DiffServ handles QoS based on per-hop behavior (PHB) as illustrated in Figure 4. Only edge routers manage QoS. This per-hop or network cloud needs to be concatenated with the next network cloud or per-hop traffic. Therefore, a distributed call admission control or a Bandwidth broker is needed for this concatenation.

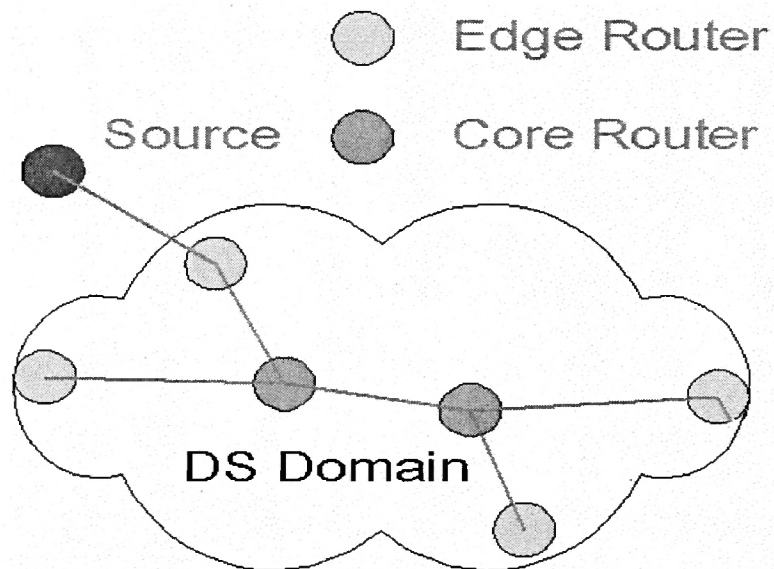


Figure 4: DiffServ as a per-hop behavior

4.2 DiffServ QoS architecture's components

Figure 5 illustrates the packet processing plan components [25]. The packet plan includes classification, policing, marking, and queuing and scheduling. The policing assures that the traffic level is within the QoS contract.

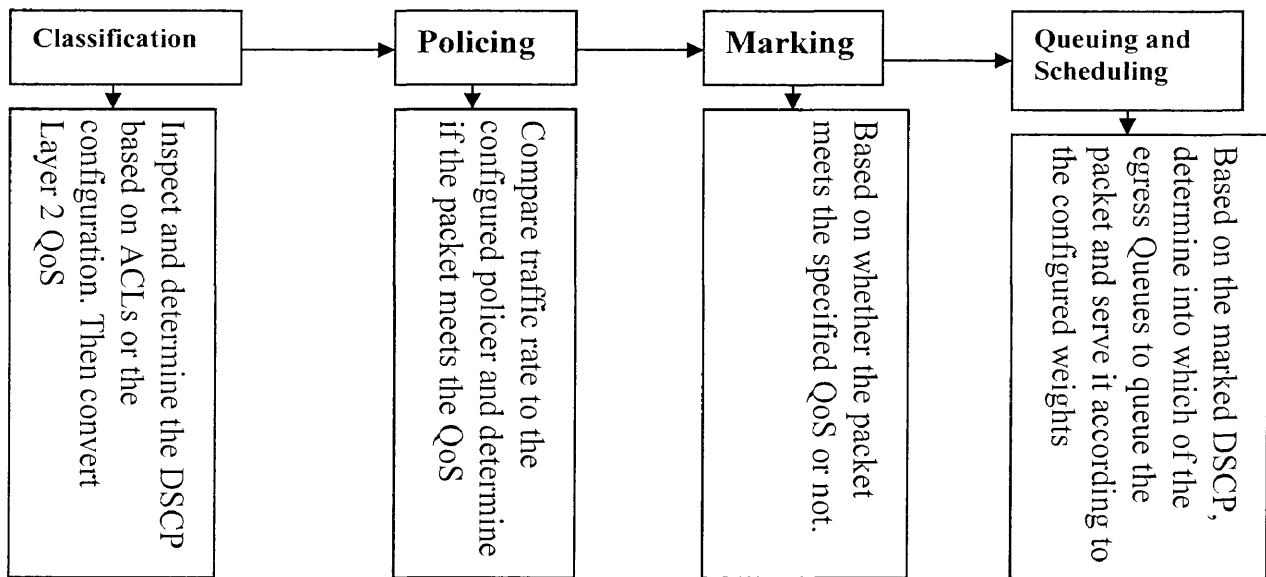


Figure 5: DiffServ Packet plan handling

To achieve this goal, policing might drop a packet or mark the packet to a different service level in a way that the transmission delay is not affected. In contrast to shaping, policing does not buffer the packets. Although policing and traffic shaping are similar in their goal of satisfying the QoS contract, shaping applies only to the outgoing interface while policing applies to both the outgoing and ingoing interfaces of the switch. Traffic shaping smoothes traffic bursts and buffers the packets that do not meet the QoS matrix.

In addition, traffic shaping affects the delay and the jitter. Shaping, policing, and scheduling are based on the ability to identify the class of service of each packet. A good implementation of the QoS policing, shaping, and marking is in Catalyst 6000 and Catalyst 4000 SE3 switches. Although these switches support these QoS modules over a wired LAN, the available technology cannot yet support policing, shaping, and marking over a VLAN yet.

4.2.1 QoS policing

Policing is to assure the specified QoS. Policing is set up by specifying the QoS matrix and applies it to the switch ports. These ports are called QoS aware ports. ATM Policing is called Usage Parameter Control (UPC) if it is done at the UNI layer and Network Parameter Control if it is done at the NNI layer. ATM policing does not delay or modify the characteristics of the packet but it may drop or mark the packet if it will not meet the specified QoS. Switch Policing are used to make sure that the traffics will not violate the negotiated QoS parameters at the connection establishment. If a packet violates these parameters, it will be either dropped or marked to be treated as a lower- level class. ATM policing uses a leaky-bucket algorithm.

4.2.2 Packet classification

DiffServ classifies packets based on the information appended in the packet header. ATM classifies packets implicitly by setting up a proper VC/VP. In general, IP classifies packets based on using one or more of the IP or TCP/UDP header fields. There is single-field (rate) classification and multifold (multirate) classification. The single-rate

classification picks n bits in the header as a classification key. It gives up to 2^n service classes. The multifield classification covers multiple fields in the packet header. It can be implemented hierarchically and a single packet can match more than one class. Both IPv4 and IPv6 packets have a header field that can be used for QoS classification. This header field is called Type-of-Service (ToS) octet in IPv4 and Traffic-Class (TC) octet in IPv6. This field can be used as a DiffServ field or it is called DiffServ Code Point (DSCP). DSCP is 6 bits and it allows up to 64 different queuing/scheduling treatments for packets at routers.

Network security and payload encryption are two main challenges for packets classification. In secured networks, authentication is used to determine if a specific user is entitled to a specific class of service. Payload encryption makes it hard for a router to classify packets.

4.2.3 Packet marking

Packet marking means decreasing the packet priority instead of dropping it in case a packet is out of profile. In other words, marking allows changing the QoS level of the packet based on packet classification or policing. DiffServ uses coloring to mark packets. It changes the packet's DSCP to a color that matches a lower QoS class. In ATM, instead of dropping the cell which is out of profile, marking changes the Cell Loss Priority (CLP) to a lower one.

4.2.4 QoS-aware scheduling

Packet scheduling is the last stage for a QoS-aware packet in a network cloud. The goal of any scheduling algorithm is to maximize the throughput and achieve fairness among calls. Different scheduling algorithms have been discussed in [29, 30]. In [29] a scheduling algorithm to guarantee a specific delay rate over a wireless network through bandwidth reservation is introduced. The algorithm, which uses the earliest deadline, regulates the number of admitted calls based on per-hop reservation according to the long-term transmission characteristics of the incoming traffics. In other words, the call is admitted if the deadline requirements of incoming flows can be met with a high probability. In [30], the scheduling algorithm assigns weights to the calls dynamically such that the weights depend on the congestion levels in the neighboring cells.

A fair scheduling algorithm is an important issue when multiple users share a wireless channel. [31, 32] discuss different scheduling algorithms. Fair scheduling allocates the resources based on the weight of the packet. In [31] a fair scheduling algorithm over wireless LAN is introduced. The algorithm does not use a centralized coordinator to arbitrate medium access. An example of multi-hop scheduling is introduced in [32]. A Maxmin fair scheduling over wireless links has been discussed in [33] which treat all flows equally.

4.2.5 Traffic shaping

In traffic shaping, edge nodes may change traffic characteristics to create packet flows that meet the contracted traffic descriptor. Shaping is not a must to guarantee the QoS. Packets are subject to reshaping between edge nodes in case they are jittered at the core routers.

4.3 Bandwidth broker-based QoS

Bandwidth broker [34 and 35] is a logical entity that has been designed as a complement of DiffServ. It works as a centralized model to manage available resources among network clouds.

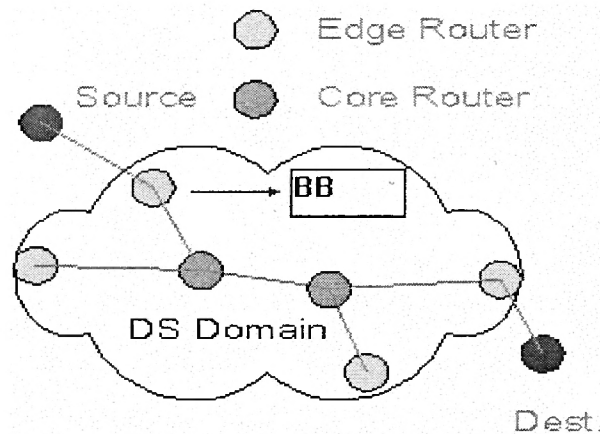


Figure 6: Bandwidth Broker based DiffServ QoS architecture

4.3.1 Bandwidth broker's features:

Decoupling packet plan from control plan: DiffServ decouples the data plan from the control plan in (per-hop behavior) PHB. However, BB concatenates this PHB decoupling process to make it end-to-end behavior. Therefore, it provides more scalable QoS by relieving core routers from QoS control management. Examples of these control functions are call admission control and QoS state maintenance.

However, handling QoS management in a centralized BB might lead to a bottleneck and lead to a scalability problem. For example, to support VoIP, applications require admission control and resource sharing. In this case a BB broker might be a bottleneck for accepting new flows while the network is underutilized.

Bandwidth broker and admission control: The BB manages resources in the network and works as an admission control among network clouds. It performs internal and external admission control. In other words, it decides whether to accept the incoming call. Therefore, it is important to provide reliable QoS. BB is a logical entity that works on a specific network domain to administer resources. The idea is to use a centralized core router to allocate the network resources and manage the QoS. Specifically, BB can dynamically create Virtual Leased Line (VLL) on demand from point-to-point in a specific domain. There are some implementations to the BB idea in [35].

Instead of having a distributed CAC algorithm through all core routers, a centralized BB could be used to merge IntServ with DiffServ.

QoS scalability: although BB has been designed to improve QoS scalability, its performance might be a bottle-neck for this scalability. Specifically, this problem might arise in case of administering a large number of traffics. For example, to handle VoIP, instead of BB, a dynamic QoS scheme (such as a dynamic CAC) is required (or dynamically provisioning the resources among calls [34]). Moreover, most of the BB implementations have problems [35] with scaling the QoS intra-domains and they do not support interior provisioning.

Chapter 5: IntServ QoS architecture

IntServ QoS is called stat-based QoS. In order for IntServ [36] to admit a flow, it makes pre-resource reservations for this flow. That inquires those routers to maintain state information about available resources and the path for the reserved resources for each flow. This pr-flow stat management creates control overhead at edge and core routers.

5.1 IntServ features:

IntServ has been designed to provide per-flow QoS. It enables applications to choose its suitable path with specific characteristics in a controlled service delivery. Therefore, routers need a controlled mechanism to control the QoS parameters.

IntServ is a resource reservation-based QoS architecture: end-to-end resource reservation schemes (which are connection oriented), have traffic descriptors in each hop to provide the characteristics of the incoming traffics. In receiver-based resource reservations (connectionless networks), the receiver specifies the required QoS and determines the required resources. In [37] there is a scheme for a wireless ATM QoS model based on predictive dynamic bandwidth reservation based on the mobility patterns. [38] Introduce reservation schemes that divide the available bandwidth in the channel into portions, some shared and others restricted to specific traffics. In [39], a bandwidth reservation scheme based on ATM traffic parameters is introduced. Static reservation schemes waste bandwidth. However, dynamic resource reservations and reallocation can lead to efficiency in resource allocation or high over-head. Dynamic allocation of resources occurs based on the current requirements of the flows and the congestion levels [40].

Per-flow state maintenance: IntServ architecture requires end-to-end per-flow QoS maintenance. Specifically, each flow gets its specified QoS parameters.

End-to-end controlled architecture: The controlled services can take delay as an input but other kinds of service take delay as an output or a result. In other words, in controlled services, applications can control their own delay. There are two kinds of delay; transmission delay (fixed delay) and queuing delay. The chosen path or the set up mechanism is responsible for transmission delay. But the queuing delay comes from trying to guarantee the service. The queuing delay is a function of token bucket (b) and data rate (r).

IntServ and fairness: IntServ supports link-sharing services. In this case the control and management scheme controls bandwidth and shares it among flows.

IntServ and service notification: IntServ uses call admission control. Therefore, it can notify applications with admission notification (and service-denial notification as well).

IntServ and QoS scalability: IntServ is not scalable because of the following reasons: (i) per-flow state maintenance, (ii) per-flow control overhead, (iii) QoS handling is processed for all routers, (iv) pre-call setup resource reservation leads to idle resources and inefficiency in resource management.

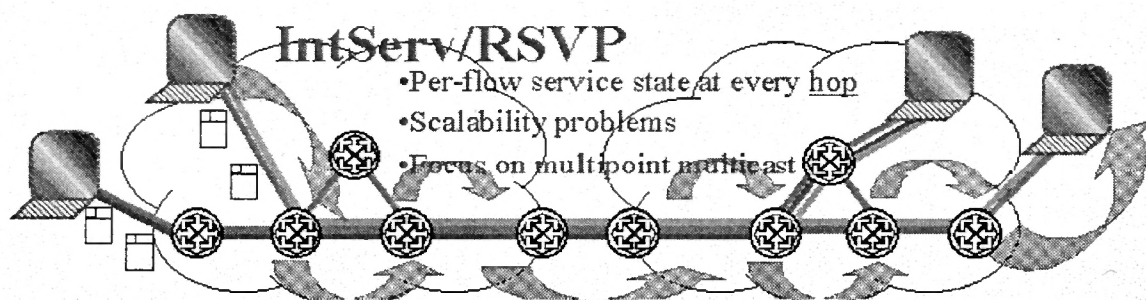


Figure 7: An IntServ model

5.2 Components of IntServ

IntServ is a controlled QoS architecture. The IntServ has four modules: a reservation protocol (RSVP), packet classifier, admission control, packet scheduler, and a control scheme.

A brief discussion about packet classifier and scheduler has been introduced in the previous chapter. The difference between DiffServ packet classifier and IntServ packet classifier is that IntServ classifier classifies packets into flows which they belong to. This classification is done based on the flow ID, source, and destination. DiffServ classifier classifies packets into class of service based on the packet ID and the class of service which have been specified at the configuration time. On the other hand, DiffServ scheduler is a PHB. In addition, it considers service classes and fairness among packets which belong to the same class. IntServ scheduler is an end-to-end behavior.

Therefore, this chapter will not address these two modules again.

RSVP and call admission control

RSVP protocol is responsible for call setup, tearing down a call, and renegotiating a call. RSVP has been designed to be integrated with many QoS control services. In other words, QoS control services are designed to be integrated with many setup mechanisms. RSVP gets the characteristics of these services and simply delivers it to routers.

RSVP carries information which can invoke QoS control service and it does not handle QoS accounting and policing.

RSVP is receiver oriented and ATM control scheme is sender oriented. In RSVP the control message is sent through a different channel (not the data forwarding channel). If either of these two channels fails, that will create a problem as long as the other one is active. RSVP allows dynamic modification for the QoS parameters but ATM does not allow any modification after initiating the VC. Instead, ATM initiates a new VC with the new characteristics and shift data transmission into it [41].

RSVP sends a path message which includes flow characteristics. If a path is available by comparing the required bandwidth with the available bandwidth, the RSVP message returns true and invokes the admission control. Admission control calculates the required bandwidth for the RSVP before RSVP send the path message. When the path message returns true, admission control reserve this bandwidth for the involved flow to prevent any other flow to accesses it.

Call admission control and QoS

The CAC concept is very important in the wireless network design for two reasons: CAC is responsible for giving a new user the access to the wireless network, and CAC is managing the network resources to maximize the utility of the network resources while guaranteeing QoS. If the resources are given to new calls regardless of the needs of the existing calls which need handoff, some of the existing calls will be dropped if the network is overloaded. Therefore, it is better to block an incoming call, which will degrade QoS, than to drop the current calls. An efficient CAC algorithm maximizes the statistical gains without violating the QoS. The efficiency of the CAC algorithm depends on how close the queuing method and the traffic estimation are to the reality [42].

One of the main goals for the CAC algorithm is achieving the efficiency in using the available resources.

5.3 Call admission control (CAC)

The concept of call admission control (CAC) is very important in network design for the following reasons: (i) CAC is responsible for giving a new user access to the wireless network; (ii) CAC manages network resources to maximize the utility while guaranteeing the QoS, (iii) it provides flows with rejection notification (if the resources are given to new calls regardless of the needs of the existing calls which need handoff, some existing calls will be dropped if the network is overloaded), and (iv) A CAC tries to achieve fairness of resource allocation among different calls.

Therefore, it is better to block an incoming call which will lower the QoS than to drop current calls. A CAC algorithm uses the information in the resource table about the network topology, traffic load, and QoS states in each path and node to make its decision. In adaptive call admission control there is a maximum and a minimum bandwidth required and maximum tolerable delay packet loss probability. The connection is admitted if at least the minimum required bandwidth is available.

5.3.1 Different approaches to deal with the CAC problem

Most CAC algorithms are based on two concepts, the effective bandwidth and complete sharing (CS). CS means a new call will be admitted if and only if the QoS for the current flows will not be violated [43] and they can be offline or online-based estimations [44]. A CAC algorithm helps the network to avoid congestions.

The CAC algorithms vary based on the queuing method, the scheduling algorithm, the buffer size, the time sensitivity of the service, and the traffic load estimation method. However, there are two main categories of CAC algorithms. The first category is parameter-based admission control algorithms; the second category is measurement-based admission control algorithms. There are per-hop CAC algorithms, clustering-based CAC, and end-to-end CAC

Service classification-based CAC algorithms are the most suitable for multimedia applications. Handoff-based CAC is dealing with one of the wireless challenges. End-to-end CAC is suitable for wired networks (not wireless because it is not scalable).

Handoff Based Call admission control algorithm:

Most of the CAC algorithms use the tuple of parameters (PCR, SCR, MBS) for both the current and the new call to make the call admission decision. PCR is the peak cell rate, SCR is the sustainable cell rate, and MBS is the maximum burst size.

Solutions for Admission Control Problem

- 1- Resource reservation approach: in which resources are reserved for the handoff calls and the rest of the resource could be used for admitting new calls.
- 2- Adaptive Admission Control: in this approach calls are admitted based on the load in the neighboring cells. If the neighboring cells have high load, few calls get admitted. If the neighboring cells have low load, many calls get admitted.
- 3- Early Reservation Approach: in this approach, resource reservation is done based on some information. Some of this information is the position of the mobile host, its direction of movement. Based on this information resources are reserved only in the target cell. This approach has a lot of control overhead which will decrease the effective bandwidth.
- 4- Probabilistic Resource Estimation and Semi Reservation Scheme: in this approach we reserve resource in the neighbor cell based on the probability for the hot mobile to move to this cell [45].

5.3.1.1 Parameters-based Admission Control Algorithms (PBACA)

Parameters-based admission control algorithms are analyzed using formal methods. In parameter based admission control algorithms, traffic parameters are estimated based on the characteristics of the expected incoming flows. Based on that estimated parameters, the network tries to estimate the needed resources. If these resources are available, the call gets admitted (if not the call is rejected).

Disadvantages of PBACA:

1- It is difficult to estimate the characteristics of the flow in advance in a multi media network.

2- The estimation of the needed resources to satisfy the specified QoS parameters is based on the worst-case analysis. Therefore, the network is underutilized all of the time.

Advantages of PBACA: The main advantage of these algorithms is their ability to provide higher QoS compared to the MBACA algorithms.

5.3.1.2 Measurement based Admission Control (MBAC)

MBAC [46] is based on estimating the QoS parameters from current actual traffics rather than using the traffic descriptor. The pre-defined traffic descriptor based CAC algorithms are static but the MBAC is dynamic. Therefore, these algorithms have two major problems. It is difficult to predict the traffic characteristics precisely. Therefore, the network resources would be either overloaded or underutilized. However, MBAC algorithm avoids such problem by taking its measurements for the existing application. Every MBAC algorithm has two components: the measurement procedure to estimate the current network load, and the algorithm, which uses this estimation to make the

admission decision. These algorithms require a traffic descriptor to describe the worst case scenario for each flow. The measurement of the traffic load is taken based on the aggregate state. On the other hand, the admission control decision is taken based on each individual state. Getting the measurements aggregately and taking the admission control decision per-flow minimizes the overhead. However, there are some other MBAC algorithms take the measurement process based on per-flow state and others have assumptions about the behavior of the flows.

Examples of MBAC

1- The Simple Sum Algorithm:

The simple sum CACA [47] is the most common admission control algorithm and it is widely used over routers and switches. Basically, this algorithm ensures that the traffic load does not exceed the link capacity.

2- Measured Sum:

In the measured sum algorithm, the call is accepted if the needed bandwidth for the incoming call plus the current load (the occupied resources) is less than the link or the channel capacity. The link fails when it has very high utilization. So, the Measured Sum algorithm specifies a utilization goal which is usually less than the link capacity.

3- Measured Admission Control Algorithm:

In this algorithm, a flow get admitted if the sum of the peak rate of the flow plus the estimated bandwidth of existing flows is less than the link capacity. The estimated bandwidth takes consideration of the packet loss rate.

4- Hoeffding Bounds (HB):

This admission control algorithm computes the equivalent bandwidth for a set of flows using the Hoeffding bounds. A flow is admitted if the peak rate of this flow plus the measured equivalent bandwidth is less than the link capacity. An exponential measurement process is used to estimate the network load.

Advantages of MBAC

- 1- It maintains higher utilization for the network.
- 2- The QoS of this model is easy to be measured because it depends on the aggregate behavior of the flows not just one individual flow.

Disadvantages of MBAC

- 1- Estimating the network load does not include the admitted flow at the measurement point.
- 2- The estimations are made based on arbitrary targets such as the utilization goal in the measurement sum algorithm. Non proper target could lead to network under-utilization or over-utilization.

5.3.1.3 Other categories of call admission control algorithm

- **Handoff based CAC algorithms**

There is an evaluation for programmable handoff techniques in [5]. The first technique is a multi-handoff access network service, which can support simultaneous handoff control over the same physical wireless infrastructure. The second one is a reflective handoff service, which allows mobile hosts to roam freely between different wireless networks.

The evaluation results say that programmable handoff techniques are scalable to support a large number of mobile hosts without side effects on the QoS.

In addition, we can reuse the frequency channels [48] to increase the network resources. Increasing the available resources will decrease both new call blocking and the handoff blocking rates. Therefore, the QoS parameters, specialty bandwidth and reliability, will be improved.

In [49] a call admission mechanism is introduced based on the sensitivity of the applications to the various parameters of QoS. Simply, this approach gives higher priority to the video and audio applications in admission and gives the data applications lower priority. This priority allocation process is based on the fact that data applications are more sensitive to cell loss than delay, and video applications are sensitive to delay more than cell loss.

In [50] there is an evaluation for programmable handoff techniques. The first technique is a multi-handoff access network service, which can support simultaneous handoff control over the same physical wireless infrastructure. The second one is a reflective handoff service, which allows mobile hosts to roam freely between different wireless networks. The evaluation results say that programmable handoff techniques are scalable to support a large number of mobile hosts without side-degrading QoS. Giving handoff higher priority than new calls has been studied in [51].

Using guard channels with buffering has also been studied in [52]. In the guard channel schemes [53] a higher priority is given to handoff handling and some resources are reserved for handoff calls. The guard channels technique has been introduced to reduce the handoff blocking rate. In [54], a CAC and bandwidth reservation scheme in wireless cellular networks has been introduced. This scheme depends on using the learning theory and statistical information to predict when and where the user will be mobile. Call admission decision and bandwidth reservation are made based on this prediction. Simulations show a decrease in handoff dropping rate and an increase of resource utilization.

- **DiffServ and Service classification based CAC algorithm**

Giving priorities and classifying services are very important in a resource poor environment. In [49] a call admission mechanism is introduced based on applications' sensitivity to various QoS parameters. Simply, this approach gives higher priority to the video and audio applications in admission and gives the data applications lower priority. This priority allocation process is based on the fact that data applications are sensitive to the cell loss more than delay and video applications are sensitive to delay more than cell loss.

There is an ATM-based CAC algorithm [55]. This algorithm classifies flows into three classes. Each class is given a specific priority. If there are not enough resources for a specific flow, this flow is buffered until the needed resources become available or the time outs. The main goal for this algorithm is maintaining the current calls and admitting

the higher-priority calls. This algorithm is a good algorithm for a scalable wireless networks and it provides a good scheme for QoS maintenance.

In [56] services have been classified into three categories and given priorities based on this classification. Based on this priority, channels and buffer capacity are being allocated to the services. [57] Introduce a service-class-based call admission control. Specifically, the CAC algorithm assigns a portion of bandwidth to be shared among service classes and the other portion to be assigned to a specific type of service class. In addition, the CAC schemes take into account the class of traffic, the required bandwidth by each call, the number of calls, and the available resources. They classify the traffic into real-time traffics and non real-time traffics.

If the source sends multi media applications, buffer sharing or space-priority mechanisms could be used in CAC. If each source has a specific traffic class, then it is better to provide a buffer with a different size for each class. For example, assigning a small buffer size for real-time traffics and a large buffer size for data traffics and giving the real-time buffer higher priority than the data traffic buffer [58].

Summery for call admission control categories :**- Cell-loss based CAC algorithms**

This category includes;

- Equivalent bandwidth (effective bandwidth)
- Upper bounds for the cell loss probability.
- Diffusion approximation.

Cell-delay based CAC:

These algorithms are scheduling-dependent algorithms, this category includes

- weighted fair Queuing (WFQ) or Packet-by-Packet Generalized Processor Sharing (PGPS) scheduling.
- Static priority (SP) scheduling.
- Delay-Earliest Deadline first (EDF) scheduling.

5.4 ATM based QoS

ATM has been designed as a QoS aware technology. It supports of-line QoS negotiation through VP/VC setup. It has the ability to provide different bit rates or bandwidth on demand. A lot of work has been done based on this concept to enhance the QoS. The QoS over ATM has been discussed in [59, 60]

Regarding the CAC, VBR makes it difficult to get an efficient CAC algorithm based on a changeable parameter. But VBR provides statistical multiplexing to the flows and dynamic bandwidth allocation among these flows.

CBR provides fixed bandwidth. In CBR services bandwidth is allocated based on the peak rate. A call is admitted if the peak rate for this call plus the peak rate for the current calls is less than or equal to the channel capacity through the entire path. Therefore only hard guarantees could be obtained from these services. Therefore, the CBR allocation leads to underutilizing the network utilization. In contrast VBR maximizes the utilization of the resources by its statistical multiplexing gains. These statistical multiplexing gains (which are the reduction in bandwidth required under VBR instead of using CBR) reduce the required bandwidth for the flow. However, VBR needs a complex admission control scheme and causes more control overhead. (To get both gains from CBR and VBR without complexity and with less overhead), some researchers [61] explored the idea of smoothing the incoming data stream by using buffering capabilities to get smooth scheduling for compressed flows. These algorithms try to guarantee that the sender and receiver buffers are not overflows or underflows.

In [62] an implicational programming interface that allows applications to specify and renegotiate their QoS during the call is introduced.

In ATM, CAC determines how much resources are required for a new virtual channel (VC) and determines whether this resource is available. The goal is to maximize network utilization and guarantee the QoS for all VCs. Most CAC algorithms use the equivalent Bandwidth (EB) approach to guarantee QoS. CAC calculates the EB required by the VC for a given flow and reserves it from end to end. Available Bit Rate (ABR) and unspecified bit rate (UBR) can support best-effort delivery.

ATM provides the best QoS and the best bandwidth sharing. In the case of CBR services (such as audio and video) bandwidth is determined at the time of connection. The VBR users declare their peak rate and sustainable bit rate. ATM supports both VBR real-time and non real-time applications. The bandwidth which is not used by CBR and VBR is called AVR which is shared by non delay-sensitive applications. UBR is for applications that do not require a specific bandwidth.

During the VC set-up a peak cell rate and a minimum cell rate are negotiated. The connection is admitted only if the minimum cell rate can be provided through the entire path. In [63] an adaptive buffering congestion control scheme (QoS control) is introduced to decrease the cell-loss rate in a wireless ATM.

5.5 DiffServ vs. IntServ

Evaluation to DiffServ architecture [64]

Table 3 shows the similarities and the differences between DiffServ and IntServ. Figure 8 shows the components of both. The current internet service is precedence-based differentiated service, in which, a new flow will be accepted even if it will lead to service degradation for the current connections.

If a specific application is sent to multiple users it should be multicasted. The existing work on DiffServ does not consider the externalities coming from multicasting. Another challenge for DiffServ is concatenating the network domains in a unified end-to-end network. The DiffServ is a PHB. Therefore, Bandwidth broker concatenates and map the packets treatment and the service-level description from hop to the next hop.

IntServ vs. DiffServ components

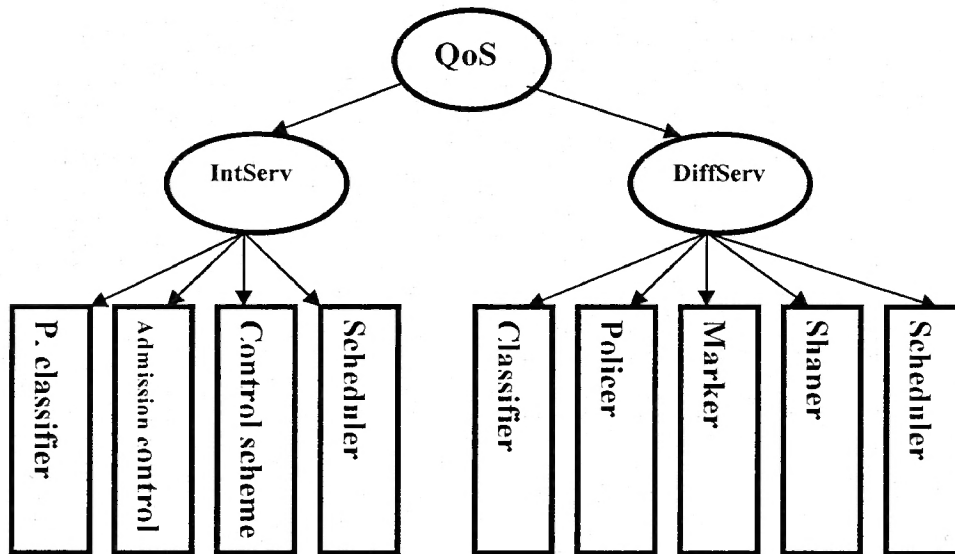


Figure 8: IntServ and DiffServ components

Features	IntServ QoS	DiffServ QoS
	Resource reservation based*	Priority and flow differentiation based*
Scalability	No scalability because of the per-flow control overhead*	It is scalable because QoS handling is done only at the edge routers
Existence	It is only used for IP and 802.*	It is been used widely
Fairness	Control scheme could be used to achieve fairness	There is no fairness in resource allocation*
Interaction with flows	It has admission and rejection notification	It does not have admission or rejection notification*
Guarantee	It guarantees QoS guarantee but no admission guarantee	no QoS guarantee
Input/output	It can take QoS parameters as input	QoS parameters are jut output
Data/control plan	It combines data plan with control plan(explicit control plan)	Decouples data plan from control plan(implicit control plan)
Resource management	It manages resources*	Does not manage resources
Delay	It controls transmission delay and queuing delay	It controls queuing delay only
Link sharing	It supports link sharing	It does not support link sharing
Utilization	It provides full utilization	Provides full resource utilization
Service support	Multimedia applications	Premium service and best effort
State	Per-flow state	Stateless
Components	Classifier, CAC, control scheme, and scheduler*	Classifier, policer, marker, shaper, scheduler*
Granularity	It does not support granularity	Higher layers supports granularity*
Domain support	En-to-end	Per-hope behavior*
Congestion awareness	No congestion awareness	It is congestion aware
Multicasting	It supports multicasting	It does not support multicasting*
Connection	It is connection oriented	It is connectionless oriented
QoS renegotiation(QoSr)	It could support QoSr	It does not support QoSr
	Service isolation	
	Absolute QoS	

Table 3: Comparison between IntServ and DiffServ QoS architecture (* indicates area of research)

Chapter 6: Previous work on adaptive QoS frameworks

IP networks are still best-effort delivery networks. In other words, the network makes no efforts to differentiate among traffic streams. However, there are two well known QoS architectures over IP IntServ and DiffServ. IntServ, which has been designed for multimedia applications, is based on resource reservation and per-flow state management. DiffServ, which has been designed to provide a scalable QoS, is based on service aggregation and differentiation.

IntServ provides better QoS guarantee for the following reasons: its per-flow state management, its admission control, and resource reservation. In addition, IntServ has end-to-end load control capability and it is a congestion aware architecture. However, IntServ is not a scalable QoS architecture (because of its excessive resource reservation and its per-flow state management).

DiffServ QoS architecture is a scalable architecture because: it is a per-hop behavior and handles QoS management only on the edge routers and it differentiates traffics and aggregates them into a small number of service classes. This service aggregation decreases control overhead compared to per-flow state management. Moreover, DiffServ impedes the control function in the packet header which decreases the control overhead as well.

Being a per-hop behavior, DiffServ does not provide an end-to-end QoS or end-to-end load control. In addition, DiffServ does not have call-admission or rejection-notification

capability. Admission and rejection notification are very important for customer satisfaction and service pricing.

Neither IntServ nor DiffServ has QoS renegotiation features. QoS renegotiation process enables services to respond to network conditions and path characteristics during connection time. In this approach, services or applications are informed of network conditions through a feedback mechanism. A feedback might capture high packet loss, congestion level, packet delay or anything of interest such as service denial which we introduce in this paper. This feedback information invokes a proper service response.

This service response is a mechanism which adjusts applications' requirements to adapt with network conditions. For example, a network node might drop lower priority packets as a response to system feedback. Most of the applications have non-negotiable QoS parameters and negotiable QoS parameters. For example, data applications can renegotiate packet delay and jitter but they cannot renegotiate cell loss rate (CLR). Audio and video applications can renegotiate required bandwidth, CLR, and packet delay but not jitter. In other words, Video and audio applications can tolerate higher delay and CLR. In addition, they can adapt to the available resources if they can be informed at the connection time. An example for service response is: encoding schemes could be used to make voice applications adaptable for transmission-rate range between 8kbps to 128kbps. In addition, Video applications could be encoded to tolerate bandwidth range between 29kbps to 1500kbps.

Obviously, the QoS renegotiation architecture has three integrated components as shown in Figure 9. The first is a feedback mechanism to provide applications with information about network conditions and path characteristics. The second is a service-response scheme to adapt with network conditions and path characteristics. The third is a load-control scheme to estimate available resources or to adjust utilization target if needed. In addition, applications should be able to tolerate lower QoS.

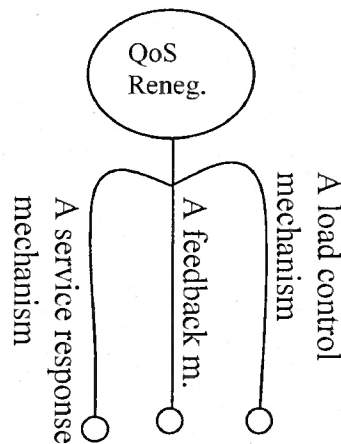


Figure 9: the components of the QoS renegotiation architecture

Dynamic QoS renegotiation has several advantages over static-resource reservation schemes. Specifically, it provides a better system performance and a better guaranteed QoS. QoS renegotiation increases the vertical scalability of QoS by emphasizing the sharing concept among calls and decreasing call drop rate.

Adaptive QoS emphasizes the concept of resource sharing among flows, which improves fairness. In other words, the system is trying to be as fair as possible by asking flows to renegotiate their QoS requirements instead of being rejected. In addition, it decreases the probability of service denial either before the connection set-up or during the connection. As shown in Equation 2, QoS renegotiation is required when the bandwidth of the incoming flows is greater than the bandwidth of the outgoing flows.

Eventually, service revenue will increase by improving the system performance and the QoS. .

$$\text{Equation 2: } \sum_{i=1}^{i=n} in - flowBW > \sum_{i=1}^{i=n} out - flowBW$$

6.1 Previous work

In this section, we will discuss previous work in different components of renegotiated QoS architecture. As shown in Figure 9.

6.1.2 Feedback Mechanisms

For a good performance of adaptive multimedia applications over wireless networks, applications should get accurate dynamic estimations of the available resources for an end-to-end path. The process of getting information about the available resources and network conditions is called *feedback mechanism*. If aggregate *in-flow* is greater than *outflow*, then delay, delivered bandwidth, or drop rate can be bad.

In a feedback-based architecture, the system is trying to be fair as much as possible by asking flows to renegotiate their QoS requirements instead of being rejected.

Capturing path characteristics in advance is not an easy task. An alternate for that is using a feedback frequently to get the path characteristics. This feedback will allow applications to adjust their requirements to the current network conditions. For example, a network node might drop lower-priority packets as a response to system feedback. A feedback might capture high packet loss, congestion level, or anything of interest such as service denial. This feedback will allow applications to renegotiate their requirements to avoid service denial. This scenario achieves fairness among flows. In other words, it is better for applications to tolerate low QoS other than getting rejected or disconnected. Another example for feedback mechanisms is impeding router mobility. In this case, transmission rate should be reduced to decrease buffering requirements at the access point to avoid a high loss rate. This feedback requires nodes to be aware of the level of service an application is getting at each hop.

Feedback mechanisms are used to maintain adaptability between applications and all IOS layers. Periodically, the application and all layers get feedback information about the network conditions, especially available bandwidth, delay and different congestion levels. In a heterogeneous network such as a wireless network, applications should adapt themselves dynamically to the environment and the available resources. Based on such information, encoding schemes will be modified. Moreover, errors feedback can enable the network to modify the error-protection mechanisms.

The QoS feedback mechanisms have been discussed in [65, 66, and 67]. Coding and the error forward correction can be used in a feedback mechanism. Previous work in this area uses information about CLR, packet delay, or bandwidth adjustments as feedback information.

In [65] a feedback mechanism has been introduced in which the management of the resources is decentralized and applications themselves can trade off resources and quality according to their own strategy. In [68] a QoS model based on the packet-loss performance and effective bandwidth is introduced in wireless networks.

Ingo et al. [69] introduce a dynamic bandwidth adjustment for multimedia applications' control. It uses packet loss rate as an indicator for the congestion state to adjust bandwidth to meet the packet loss-requirements. Bolot et al. [70] proposed a scalable end-to-end feedback mechanism for video applications. They used a random deployed reply scheme from receiver to sender to identify the network state. An adaptive feedback scheme to control congestion state has been proposed in [71]. This scheme requires switches to send the transmission rate and buffer occupancy to the source. Based on this information the source adjusts its transmission rate.

CLR has been used as a feedback mechanism in [72, and 73]. These two approaches adjust link utilization to provide a specific CLR. A dynamic bandwidth adjustment mechanism has been proposed in [69]. In this approach, the network manager informs applications of network congestion levels. The sender uses feedback information from

receiver about CLR which indicates the congestion level to adjust bandwidth. The experiments for this approach have been done for video conferences

Feedback mechanisms have two benefits:

- Obtaining accurate information about the path characteristics and QoS parameters in the path will allow services to take the proper reaction and act precisely comparing to the static behavior of indirect measurements or static reservations.
- Feedback about the path characteristics allows shorter timescale reaction from applications.

Among several adaptive QoS frameworks, two well-known architectures are discussed: Seamless Wireless ATM Network (SWAN) and AQuaFWiN [7]. SWAN is an ATM wireless multimedia network developed at Bell Laboratories [74]. SWAN enhances the ATM networks to support wireless networks. Its main function is to connect heterogeneous wireless ATM networks at the end hop. It uses the MAC delay information for packets as a feedback mechanism.

In AQuaFWiN adaptive QoS framework, packet probing has been used to capture the path characteristics and give this information to the application as a feedback mechanism.

There are some concerns regarding feedback mechanisms usage:

- A good feedback mechanism should have low overhead and should be associated with a proper service response and a load-control mechanism.
- Developing a feedback mechanism in a multicasting environment is a major challenge.

6.1.2 Load Control Schemes

Load-control or bandwidth adjustment techniques are very important in the QoS renegotiation process. The control scheme or the network manager usually monitors and controls the congestion level in the network. If this network manager satisfies specific delay, cell loss, or bandwidth requirements, it is called QoS aware network manager. If this network controller manages the relationship between applications and network conditions based on feedback information between the two parties, it is called *adaptive network controller*. In [66], the control theory has been used to model QoS adaptation.

In per-flow measurements, bandwidth adjustments are done when a call is received and when a call is ended, which leads to a lot of overhead.

Using static bandwidth control to achieve QoS leads to underutilizing the available resources and may degrade QoS. Instead, adaptive bandwidth control should be used to control the queue length, the packet loss, and the packet delay. It refers to some work which use cell loss rate as a feedback to the system to adjust bandwidth in a way that control the queue size to control the cell loss rate in the end

A survey of different adaptive load-control algorithms to guarantee QoS is discussed in [75]. A bandwidth-control scheme to guarantee a bounded probabilistic delay has been introduced in [76]. It adjusts bandwidth to achieve a specific delay rate. Another bandwidth-adjustment scheme that guarantees a bounded delay has been proposed in. Some concerns [75] regarding using bandwidth adjustments to achieve QoS are:

- Which time domain should be used to adjust bandwidth target to avoid excessive overhead.
- Adjusting bandwidth target frequently could lead to much overhead and poor performance because of inaccurate feedback obtained from short-term measurements.
- Bandwidth adjustments affect other renegotiable QoS parameters such as CLR and packet delay.

6.1.3 Service response mechanisms

Coding and errors forward correction could be used as a service-response mechanism. Specifically, voice and video could be encoded differently. In other words, encoding parameters can be modified dynamically as a response to a changing environment. Encoders use different compression rates to get different QoS levels. Adaptive encoding approaches could be used to smooth the bandwidth of the encoding streams. Adaptation could be done by encoding [10], scaling [11], or filtering techniques [12]. Another example for service response mechanisms is impeding mobility-aware routers. In this case, transmission rate should be reduced to decrease buffering requirements at the access point to avoid high cell loss rate. This feedback requires nodes to be aware of the application's level of service at each hop.

6.2 Examples of adaptive QoS aware architectures

1- **Cisco Virtual Switch Architecture (CVSA):** CVSA has four modules: application control module, forwarding and adaptation, a dynamic partitioning function, and a virtual switch interface. The partitioning function allows highly granular control of individual ports to facilitate the usage of multiple control planes on a single port. This switch has different ports for delivering ATM traffics and IP traffics [77].

Its feedback mechanism requires nodes to calculate their delay and available bandwidth and route this information to the other nodes to be used especially by any measurement-based admission control scheme. This switch does not provide per-flow QoS but it provides per-link QoS estimation and a dynamic QoS matrix.

2- **Mobiware:** The COMET group at Columbia University [78] has developed an adaptive QoS-aware middleware platform called Mobiware which supports multimedia applications over wired and wireless ATM networks.

This platform has the following adaptive features:

- It has an adaptive network layer that supports QoS handoff control over seamless media.
- The adaptive application layer provides hard guarantees for minimum acceptable QoS and uses residual bandwidth to deliver enhanced QoS when resources become available.

3- **RDRN:** Kansas University has developed The Rapidly Deployable Radio (RDRN) system [79] to support wireless ATM networks. It supports an adaptive link layer to ATM cells, which are called WATM frames and are classified in a separate queue and are

handled separately. It has adaptive network layer features that support handoff. The system predicts mobility trends and its velocity based on which the system allocates resources in advance to handle handoff successfully. It uses the graph-coloring to assign frequencies among calls.

4- **SWAN**: The Seamless Wireless ATM Network (SWAN) is an ATM wireless multimedia network which has been developed at Bell Laboratories [74]. SWAN enhances the ATM networks to support wireless networks. Its main function is to connect heterogeneous wireless ATM networks at the end hop. SWAN uses the MAC delay information for packets as a feedback mechanism.

5- **WAMIS**: WAMIS [24] is a wireless project that supports adaptive applications, heterogeneous environments and multiplication services.

The adaptation features of WAMIS:

- Clustering: Enhancing the performance of the wireless networks in this architecture depends mainly on the concept of clustering. Clustering in this project provides a framework for enhanced CDMA code separation (link layer-based adaptation), power control scheme (physical layer-based adaptation), and channel reservation (network layer-adaptation). WAMIS uses graph-coloring algorithms for clustering implementation, network partitioning, and codes assigning. It uses the shortest-path algorithms as a fast-reservation scheme that supports mobility. In the data-link layer, CDMA and TDMA are combined, leading to 80 percent enhancement over TDMA.

Chapter 7: The proposed approach

The proposed approach is implemented over IntServ architecture because of its control features and calls rejection and admission features. In addition, it considers the adaptive characteristic of audio and video applications. Video and audio applications can tolerate higher delay and higher CLR. In addition, they can adapt to what available resource the system if they can be informed of that at the connection time. It classifies services into two categories guaranteed services and best-effort services. The differences between the proposed approach and the other IntServ approaches are: (i) it does not perform per-flow calculations and estimations. These estimations are done on demand as discussed below; (ii) the proposed scheme enables applications to communicate with the network concerning information about the available resources.

The algorithm makes calculations for measurements periodically or only when a call gets rejected. This happens by using time (t) as a control parameter to invoke the call admission control. This time scale varies by the congestion level in the network.

My argument for that algorithm is that if the network is facing a low level of congestion, there is no need for neither service differentiation or for measurements. Obviously, all Packets will be processed based on FCFI policy. In case of calculations either for service differentiation or for measurements (resource reservations) there is always delay which is called processing delays. This processing delay will add to the delay of the high congestion period. In other words, the processing delay is always forwarded to the next

period for each flow. This delay will add to the delay of the current packet. In addition, renegotiating QoS parameters is preferred to service denial.

The common problem of IntServ is that it is not scalable because of the control overhead which is done for each flow (even for the first flow) regardless of the congestion level of the network.

This thesis initiates a new adaptive feedback mechanism over IntServ. It allows multimedia applications to renegotiate their QoS requirements to adapt with the available resources. The proposed feedback mechanism is based on the call-rejection notification. In addition, we introduce the concept of QoS renegotiation gains and try to calculate it.

We claim that this approach improves the system performance provides a better QoS and higher service revenue. Moreover, this work proposes the first attempt to calculate the adaptive QoS gains. Eventually, it introduces a new QoS parameter which is called rejection notification and emphasizes its importance in service pricing and QoS.

The proposed call-rejection notification based feedback mechanism is implemented only at the network layer of the source node by using the resource estimator and call-admission control capability to find whether there are enough resources for the incoming flow. If there are not enough resources for this flow it is rejected and notified of this rejection and asked if it can renegotiate its QoS (which is the bandwidth in our case). Bandwidth has been used as a renegotiable QoS parameter because of its impact on other parameters such as CLR and packet delay. If its QoS parameters are renegotiable, the

admission control gives the application the minimum QoS or the renegotiable QoS parameters. In this case, call-rejection notification indicates high congestion level. Therefore, flows are asked to share the available resources and tolerate lower bandwidth for the moment.

Traffics arrival does not follow a normal distribution. In other words, traffics fluctuate during the connection. And the call-admission control unit allocates and reallocates bandwidth for each incoming and outgoing flow. Therefore, periodical resource estimation is done to update routers with the current estimation of available resources. This periodical resource estimation process is to avoid excessive resource reservation. Updating routers with information about available resources enables routers to know when they can offer renegotiable QoS or the requested QoS. We call this periodical resource estimation process associated with the *call rejection notification* feedback mechanism *On-demand resource estimation*. Choosing the time period for resource estimation is arbitrary or could be based on archival information.

$$\mathbf{Equation\ 3 : } \min \sum_{i=1}^n fBW_i \leq aBW \leq \sum_{i=1}^{i=n} flaBW_i$$

As indicated in Equation 3, QoS renegotiation occurs when the sum of the minimum required bandwidth of all flows ($\min \sum_{i=1}^n fBW_i$) is less than or equal to the available bandwidth in the link (aBW). But if the current available bandwidth in the link is greater

than or equal to the sum of the long-run average required bandwidth for all flows $(\sum_{i=1}^{i=n} fl_{aBW_i})$ the requested QoS parameters will be granted.

We have some arguments for using periodical resource estimation instead of per-flow estimations. Figure 10 shows that link utilization starts low and increases with time. Link utilization increase over time and resource estimation is needed as a result. In other words, we do not need to make estimations and measurements until the call start to get rejected or periodically just to updates routers with available resources.

Therefore, making resource estimation at this low link utilization period is excessive. That is the main reason for IntServ not to be scalable. It is obvious that there is no need for service differentiation, resource estimation, or resource reservations in this case. Obviously, all flows will be processed based on FCFI policy. In case of excessive resource reservations or even excessive service differentiation there is always a delay which we have called flow processing delays. This processing delay will add to the delay of the high-congestion period. In other words, the flow processing delay is always forwarded to the next time period for each flow. This delay will add to the delay of the current packet. We will discuss the flow-processing delay in section (8.3). In the long run, traffics fluctuate in the link. Therefore, periodical estimations are needed to distinguish the high-congestion periods from low-congestion periods as shown in Figure 10.

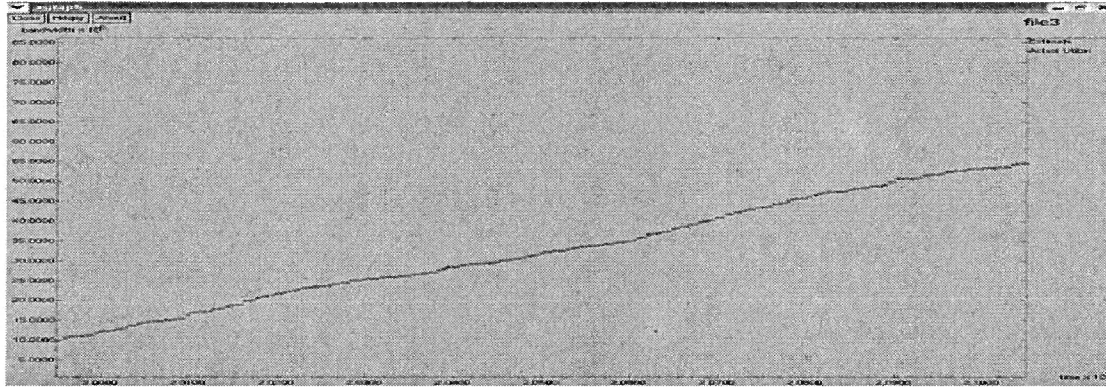


Figure 10: the link utilization trend after running simulation

Figure 11 shows the QoS renegotiation gains in the link. $L1$ represents the sum of the minimum required bandwidth for each flow. It is obvious that in $t1$ resource estimation is useless because the link has low utilization. In periods t_2 , t_4 , and t_6 , the QoS is renegotiated and flows are provided with only the minimum required bandwidth or the renegotiated bandwidth because of the high congestion level. In periods t_3 , t_5 , and t_7 flows are provide with their requested bandwidth. QoS renegotiation gains (which has been calculated in Equation 4, come from admitting more flows and decreasing the control overhead. In addition, there are indirect gains from the adaptive QoS which is improving the QoS parameters. Decreasing CLR will decrease retransmission overhead.

Equation 4 : QoS renegotiation gains = control overhead + retransmission overhead + QoS enhancement + increasing the call success rate.

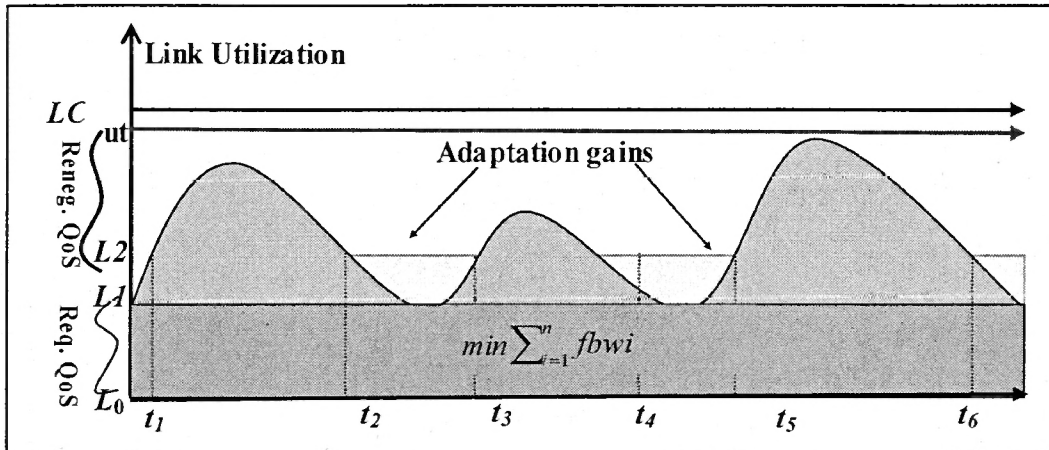


Figure 11: an adaptive link traffics with adaptive gains

Our work has some similarities and differences with AquaFWiN. The similarity is: the application's response to the available resources and its ability to renegotiate its QoS within a range of bandwidth, CLR, or packet delay rate. The differences are: our approach has been implemented over IntServ to improve its performance and make it more adaptive using QoS renegotiation. In addition, our approach uses call rejection notification as a feedback mechanism and introduces this concept as a QoS parameter as well. Moreover, we differentiate between requested QoS and renegotiated QoS. Requested QoS is the satisfactory QoS that application should obtain in normal conditions based on its request. Renegotiable QoS is the minimum tolerable QoS that application can obtain. However, the major deficiency of the previous work is ignoring to maximize service revenue, which we consider in our work, while guaranteeing QoS. It is not a good practical solution to maximize QoS parameters at the expense of service revenue.

7.1 The algorithm description and implementation details

The simulation environment and topology description:

For simulating all scenarios, we used the call admission control and IntServ module in ns-2 version 2.26. For our approach, we modified it to make it adaptive using the flow information as a feedback mechanism and periodical resource estimations. We used a single source and single destination topology which is connected by 10Mbps duplex link and propagation delay 1ms. The simulation runs for 3000 sec. The source generates an exponential on/off source with a peak rate of 64k.

We have implemented three different IntServ approaches without a feedback mechanism and our new approach with the feedback mechanism and on-demand resource estimation. Figure 12 shows how the algorithm works and how the feedback mechanism and QoS renegotiation works.

Figure 8 shows the components of IntServ which has been used as an environment.

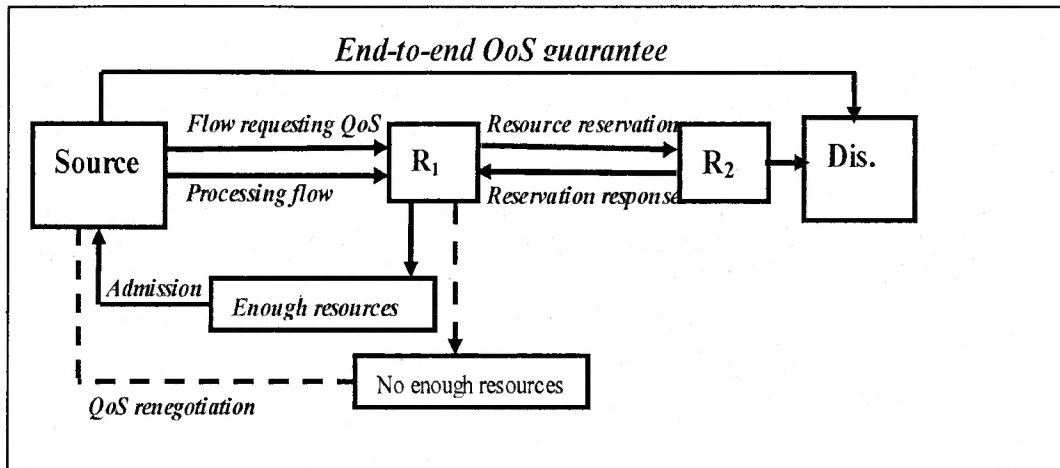


Figure 12: An adaptive QoS scheme based on resource reservation and QoS renegotiation

7.2 Call-rejection notification based feedback algorithm

General overview:

The description of how the algorithm works has been illustrated in Figure 12. The algorithm does resource estimations periodically to avoid excessive overhead. Getting this periodical information about available resources is important to enable routers to distinguish when they can support either renegotiable QoS or requested QoS.

```

1- Do periodical resource estimation ( ) {
2- Get_flow( BWi)
3- If (BWi + CurentLoad ≤ linkbandwidth)
4-     Admit (flowi)
5- Else {
6-     Reject (flowi)
7-     If rejected ( flowi )
8-         BWi = negBWi
9-     if (negBWi + CurntLoad ≤ linkbanwidth)
10-         Admit ( flowi)
11-     Else
12-         Reject (flowi) } }

```

Overview:

When a call comes to the system, the admission control module measures its bandwidth and adds it to the bandwidth of the current calls. If the sum is less than or equal to the available bandwidth, the call will be admitted; otherwise, it will be rejected. If a call has been rejected, the control module will ask the application if it can tolerate a lower bandwidth. If this negotiable bandwidth is available, the call will be admitted otherwise, it will be rejected. When the application is asked to renegotiate its QoS we decrease its required bandwidth arbitrary (e.g. 25%) assuming that it can tolerate lower bandwidth.

BW_i is the required bandwidth for flow i , $CurntLoad$ is the current traffic load estimation which has been done by call admission control, $negBW_i$ is the renegotiated bandwidth for flow i .

1- In line 1, we implement periodical resource estimation. We use t (time as a control parameter to invoke the resource estimator). In other words, our inputs are arbitrary time scale to provide the required QoS parameters. The periodical resource estimation is important to switch the system from the QoS renegotiation state to the requested QoS state after updating the routers' information about the available resources. In other words, if there are enough resources in the path, the requested QoS will be granted and the flows will not be asked to renegotiate their QoS.

2- `Get_flowi` method estimates the required bandwidth of the incoming flow i .

3- This method adds the estimated bandwidth for the incoming flow to the current load and compares the sum to the available bandwidth in the link. If the sum is less than the link bandwidth, it admits the call. The admit method reserves the bandwidth of the admitted call.

4- `Reject` method (line 7 and 8) asks the application to renegotiate its required bandwidth which we decrease it arbitrary (25%).

5- The minimum tolerable bandwidth (renegotiated bandwidth) is assigned to the flow i in line 8.

6- In line 9, the algorithm adds the new assigned bandwidth to the current flow and adds it to the current load. If the sum is less than or equal to the available resources, the

call is admitted otherwise it is rejected. If the flow is rejected, it will be queued until there are enough resources for it.

7.3 The advantages

The proposed feedback mechanism uses flow's information (not packet's information) as a feedback mechanism). This feedback mechanism does not have probing and does not involve the receiver or the lower layers like AQuaFWiN and SWAN. The proposed approach improves IntServ QoS scalability by doing periodical resource estimations. We claim that the Call-rejection notification-based feedback has low control overhead. It does QoS renegotiation during the connection. This online QoS renegotiation decreases the connection set up overhead by decreasing the call rejection rate as we will discuss in section (8.2). Using packet probing to capture the path characteristic generates overhead. In this approach, we used IntServ features to detect the congestion level by rejection notification. Using call rejection notification does not generate end-to-end probing or signaling overhead.

Eventually, the proposed scheme has two way communications between the available resources and the application through a feedback mechanism.

Chapter 8: Experimental results and simulation analysis

We claim that this algorithm increases the number of admitted calls, decreases the number of rejected calls, improves QoS, and decreases call processing time. We compare our renegotiation QoS approach with three different load-control schemes over IntServ. These three schemes are utilization target based IntServ, incoming flow bandwidth based IntServ, and IntServ based on the probability that the required resources exceed the available resources (PNREARBI). In these three approaches, utilization target, incoming flow bandwidth, and the probability that the required resources exceed the available resources are used as input parameters for load control to provide specific QoS parameters.

We have used IntServ QoS environment over ns-2 for simulation. We have chosen IntServ because it has load-control function and call admission and rejection capabilities. The simulations have been run for a single source and a single destination as shown in Figure 12. Two QoS parameters have been measured: (1) CLR to reflect the application's point of view, (2) utilization target to reflect the service provider's point of view. We run three different experiments. The first is to show the QoS improvements for the new approach. The second is to show the call success rate and call rejection rate which are important for maximizing service revenues and QoS scalability. The third experiment is to show control overhead and call processing delay to emphasize our concept of the forwarded delay.

8.1 experiment 1: improving QoS

Introduction

There are two main categories of schemes to guarantee QoS, priority-based category and resource reservation-based category. RSVP and IntServ QoS architecture belong to the resource reservation schemes.

IntServ QoS guarantees QoS by resource reservation through a call admission control, a control function, and a control variable. The control variable could be the utilization target, the bandwidth of the incoming flow, the available resource in the channel, or the probability that the required bandwidth plus the current bandwidth exceeds channel capacity. The idea behind that is assuming some parameters are constant and initialized at the configuration time and one parameter is dynamic. This parameter is called the control parameter based-on-which the network administrator can control the congestion level of the network to guarantee a specific QoS. This approach in guaranteeing QoS is similar to the way TCP works. TCP assumes that packet loss occurs because of high congestion. If TCP does not receive acknowledgement for a specific packet, it assumes that this packet has been lost because of a high congestion level. TCP resends this packet and decreases transmission rate to avoid losing more packets.

Every IntServ approach needs an estimator to estimate the current load or the bandwidth of the incoming flow or both. IntServ does per-flow calculations and the default action is to reject the flow until the QoS parameter is guaranteed. In addition to a resource estimator, IntServ requires an admission policy and both are called admission control.

8.1.1 Utilization target based IntServ QoS

This approach controls the channel utilization target (Utilization T.) to achieve the required QoS parameters cell loss rate (CLR) and link utilization (U'). In other words, if the network is facing high congestion levels, it decreases its utilization target to meet the required QoS. This approach samples the data streams to avoid unbalance in traffic flow (this is the idea behind small and equal-size cells in ATM networks).

Utilization T.	U'	CLR
98	91	8.2
95	88	4.5
92	85	0
90	84	0
88	83	0
83	78	0
85	75	0
80	75	0
78	73	0
70	65	0

Table 4: utilization target-based QoS

Table 4 and Figure 13 show that there is always a gap between the utilization target and the achieved utilization (U'). In addition, increasing the channel utilization is at the

expense of CLR. In addition, when the utilization target increases, the actual utilization increases at lower rate and CLR increases at higher rate.

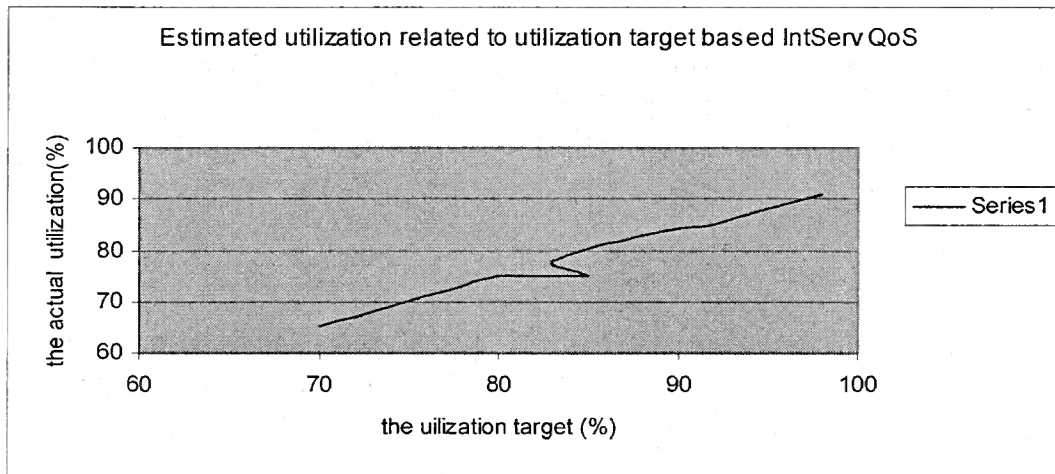


Figure 13: Utilization target-based IntServ QoS performance

8.1.2 IntServ QoS based on the probability that the needed resources exceed the available resources

This approach [80] is a static approach. It does not support QoS renegotiation and it does not get benefits from adaptation gains. It uses the probability that the needed resources exceed the available resources ($P(\%)$) to control the load to provide specific QoS parameters. Figure 14 and table 5 show the simulation results for this approach.

P (%)	CLR	U'
20	0	92
15	0	92
14	8.3	92
12.5	7.5	92
11	6.2	91
10	4.8	91
9	3.2	91
8	2.6	90
5	9.2	88
1	0	70

Table 5: IntServ QoS based on the probability that the needed resources exceeds the available resources

In this approach, the admission decision for a new call depends completely on the probability that the needed resources exceed the available resources. This approach is not robust because it depends on probability analysis. In addition, we cannot consider all available resources because we must keep a window for burst traffics. This is illustrated by the fixed channel utilization (92%) regardless of this probability. Moreover, calculation of this probability for each call implies a lot of overhead.

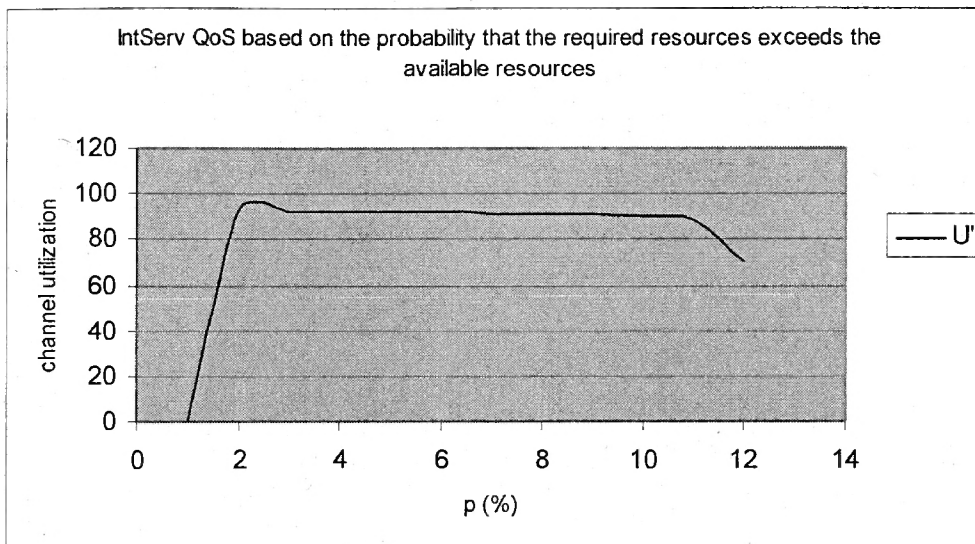


Figure 14: IntServ QoS based on the probability that the traffics need more resources that what is available.

8.1.3. The Incoming flow bandwidth based IntServ QoS

This approach [81] is a typical example of measurement-based call admission control. Parameter-based call admission control schemes and static QoS architectures assume that the incoming flow's parameters are fixed. It uses the expectation of the incoming-flow's bandwidth (R (KB) in kilo bytes) to control traffic load. Pre-estimation for the incoming flow parameters is difficult. In addition, this approach assumes fixed channel utilization (88 % as shown in Table 6) which leads to inefficiency in resource utilization. In addition, increasing the incoming-flow bandwidth increases CLR as shown in Figure 15 and Table 6. Using the available resources inefficiently makes the scalability of IntServ worse.

R (KB)	CLR	U
0.06	1.2	0.88
1	1.15	0.88
5.06	0	0.88
16	6.5	0.88
39.06	1.3	0.88
81	4.2	0.88
150.06	4.1	0.88
256	7.9	0.88
410.06	7.3	0.88
625	5.4	0.88
915.06	5.4	0.88

Table 6: The incoming flow bandwidth based IntServ QoS

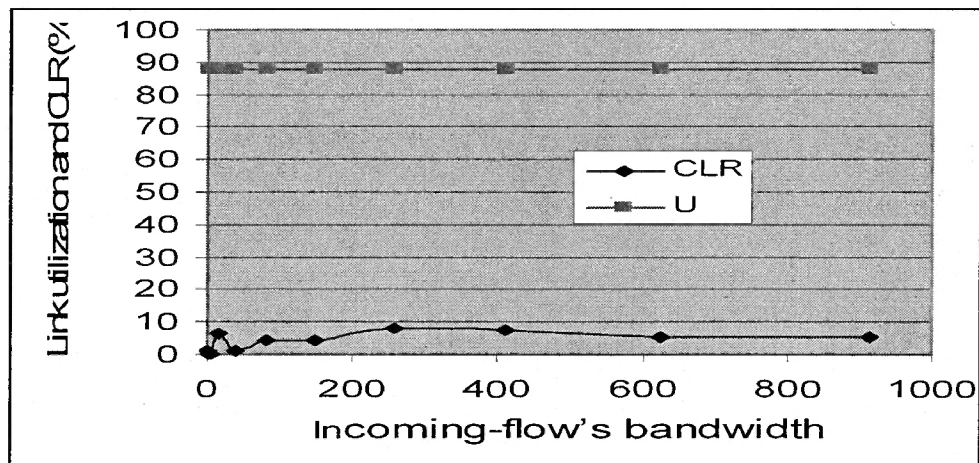


Figure 15 : QoS parameters for incoming-flow-bandwidth based IntServ

8.1.4 QoS renegotiation based on call rejection notification feedback

In this approach we used time intervals (T in sec.) as a control parameter. Table 7 and Figure 16 show the QoS parameters for the proposed approach. If resource estimation has been done every second, the CLR is 0.0004 %. That means the estimation is very accurate and it represents the link characteristics. However, the link utilization is very high (93 %) but that might include high control overhead because of the excessive resource estimation. In addition, if resource estimation is done every 3 seconds, the system provides 4.6 % CLR and 89 % link utilization.

T (sec.)	CLR	Link utilization (%)
1	0.0004	93
2	3.6	90
3	4.6	89
4	0	88
5	0	87
6	0	86

Table 7: QoS parameters for the proposed approach

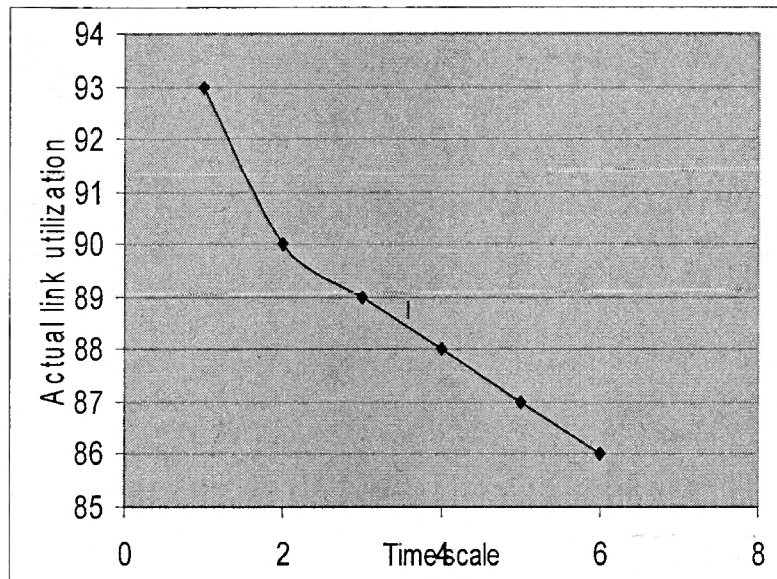


Figure 16: Link utilization with respect to time for the proposed approach

8.1.5 Comparison study

Figure 17 and Table 8 show that the QoS renegotiation approach provides better QoS parameters than the other approaches. It provides the least CLR and the highest link utilization on average for the simulation sample.

1	Average (%)	CLR	Link Util.
2	Uti. T. B. load control	2.5	79
3	PNREARB IntServ	3.5	87
4	Incoming Flow B. IntServ	4.3	88
5	Renegotiation QoS	1.2	91

Table 8: QoS parameters for IntServ different approaches comparing with the proposed approach

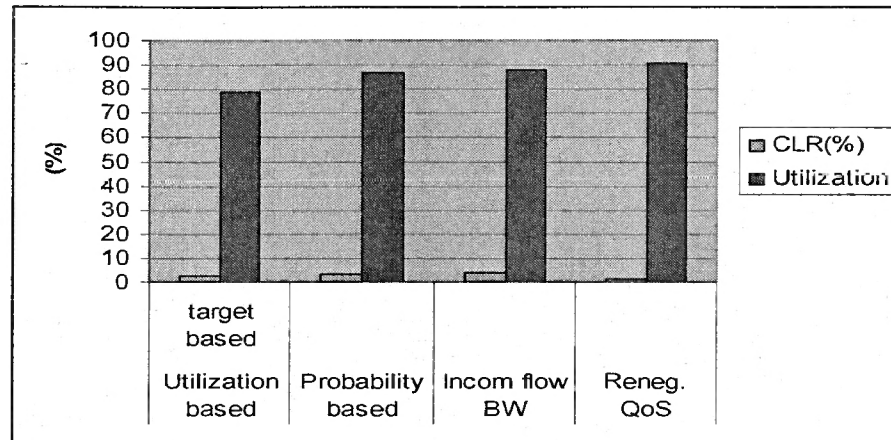


Figure 17: comparison among average CLR for the different IntServ approaches and the adaptive approach

Line 2 in Table 8 shows that the utilization target-based IntServ provides 2.5% CLR and 79% link utilization. Line 3 shows that IntServ based on PNREAB provides 3.5% CLR and 87% link utilization. IntServ based on the bandwidth of the incoming flow (line 4) provides 4.3% CLR and 88% link utilization. Line 5 shows that the proposed approach provides 1.2% CLR and 91% link utilization.

8.2 Experiment 2: Call success versus call rejection ratio

A QoS-aware network should not only be able to provide the requested QoS, but also should be able to reject services and notify these services with this rejection. In addition, improving system QoS should not be at the expense of service revenue (the ratio of processed calls)

In this experiment and the next one we have not simulated the incoming flow bandwidth because it is difficult to predict the bandwidth of the incoming call. Most of the QoS

work which has been done concentrates on enhancing the QoS parameters regardless of the control overhead and the service's revenue. In our work, we consider the service revenue (in this experiment) and the control overhead in the next experiment) while improving QoS parameters.

8.2.1 Link utilization-target-based IntServ

Table 9 and Figure 18 show the call admission and the call rejection ratio with respect to link-utilization-target. Increasing link-utilization-target decreases the number of admitted calls and increases the number of rejected calls.

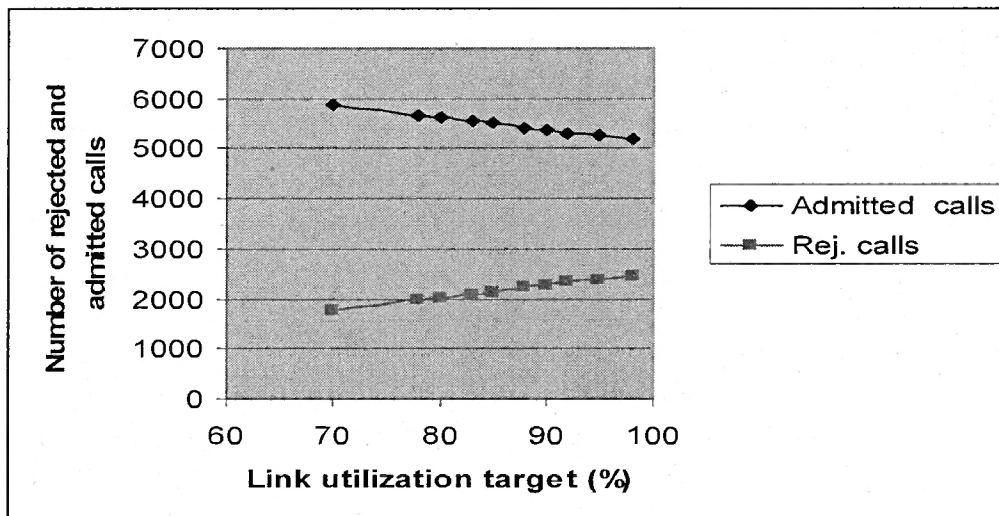


Figure 18 : call admission and rejection for Utilization target based IntServ

Util. T.	Admitted calls	Rej. calls
98	5181	2466
95	5265	2383
92	5319	2329
90	5385	2263
88	5423	2225
85	5503	2145
83	5574	2094
80	5636	2012
78	5666	1982
70	5891	1767
86(Ave)	5484 (Ave.)	2166 (Ave)

Table 9 : call admission and rejection for Utilization target based IntServ

8.2.2 The probability that the needed resourced exceeds the available resources

Table 10 and Figure 19 show call admission and rejection number with respect to PNREAR. The simulation results show that **increasing PNREAR or decreasing it does not have big influence on the number of rejected or admitted calls.**

P (%)	Admitted calls	Rejected calls
20	3272	1292
15	3211	1319
14	3244	1293
12.5	3256	1333
11	3213	1341
10	3237	1334
9	3921	1302
8	3338	1286
5	3247	1356
1	3237	1324
10.55 (Ave)	3317 (Ave)	1318 (Ave)

**Table 10 : call rejection and admission with respect to the probability that the
needed resource will exceed the available resources**

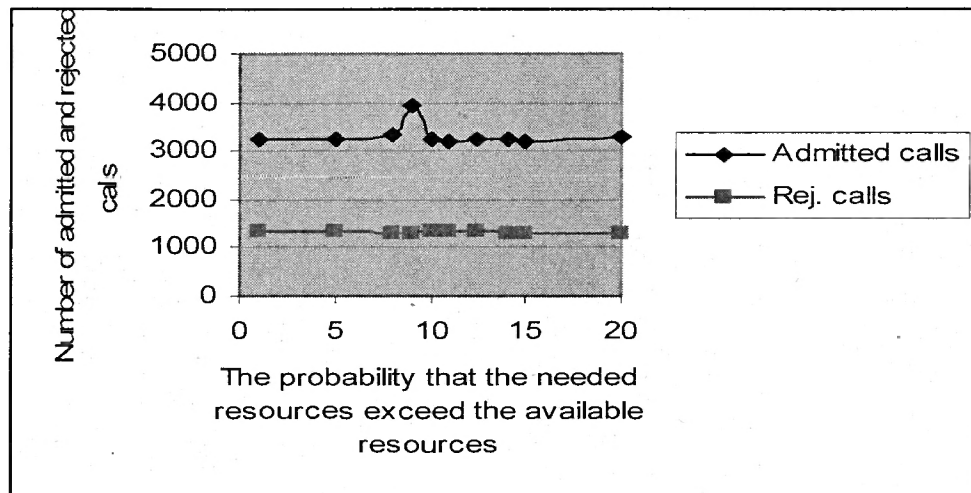


Figure 19 : call rejection and admission with respect to the probability that the needed resource will exceed the available resources

8.2.3 Call admission and rejection for the proposed approach

Table 11 and Figure 20 show that if we do periodical resource estimations associated with QoS renegotiation, the number of admitted calls will increase (the number of rejected calls will decrease as well) for big interval times. That is because using periodical resource estimation avoids excessive resource estimation and QoS renegotiation improves the system performance. Increasing the number of admitted calls and decreasing the number of rejected calls improves service's revenue.

Time (Sec.)	Admitted calls	Rejected calls
1	4473	3175
2	4742	2906
3	5293	2355
4	6414	2355
5	7180	468
6	7300	348
7	7389	259
8	7463	185
9	7463	185
10	7463	185
5.5	6518	1242

Table 11 : Number of admitted and rejected calls for the proposed approach

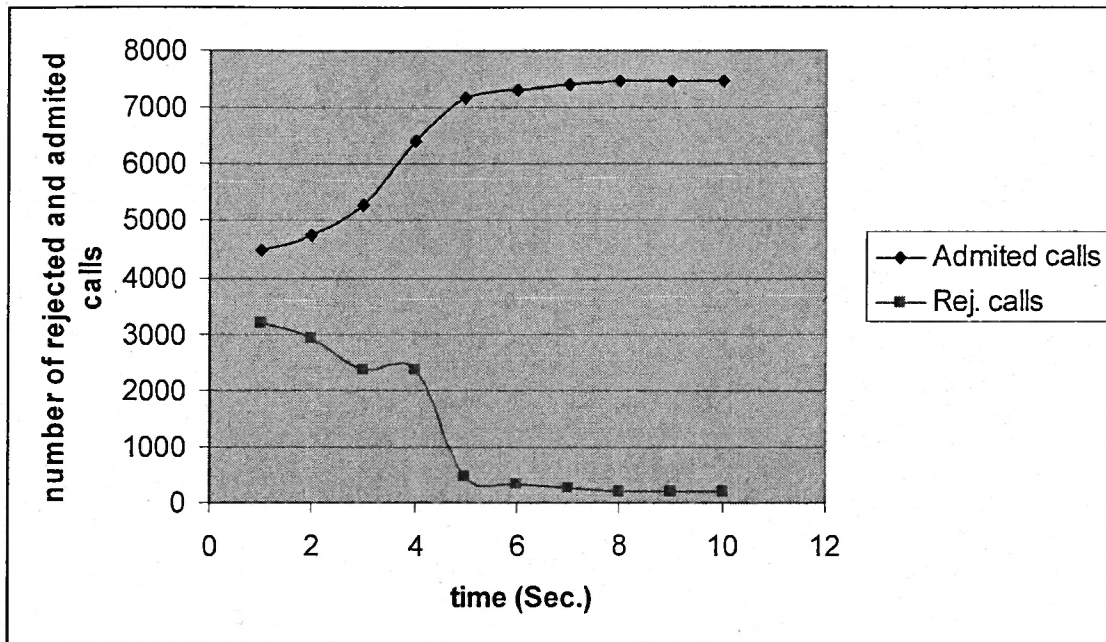


Figure 20 : Number of admitted and rejected calls for the proposed approach

8.3 Experiments 3: Call set up and per-flow processing time

When a call is rejected, it keeps waiting and tries to get access to the media until there are available resources for this call. This process involves CPU, memory, and time delay.

We define the processing delay for a flow as the time between the arrival time and the call end. We assume that all flows have the same load and the same time duration.

8.3.1 Utilization target-based IntServ

Util. T.	F. P.T.
98	164
95	158
92	152
90	148
88	144
85	142
83	138
80	133
78	129
70	114
AVE	142

Table 12 : Flow processing time with respect to link utilization

Table 14 and Figure 21 show that the average per-flow processing time (F. P. T.) is increasing with respect to utilization target. Theoretically, if the link utilization target increases, there will be high competition among flows to have access to the media.

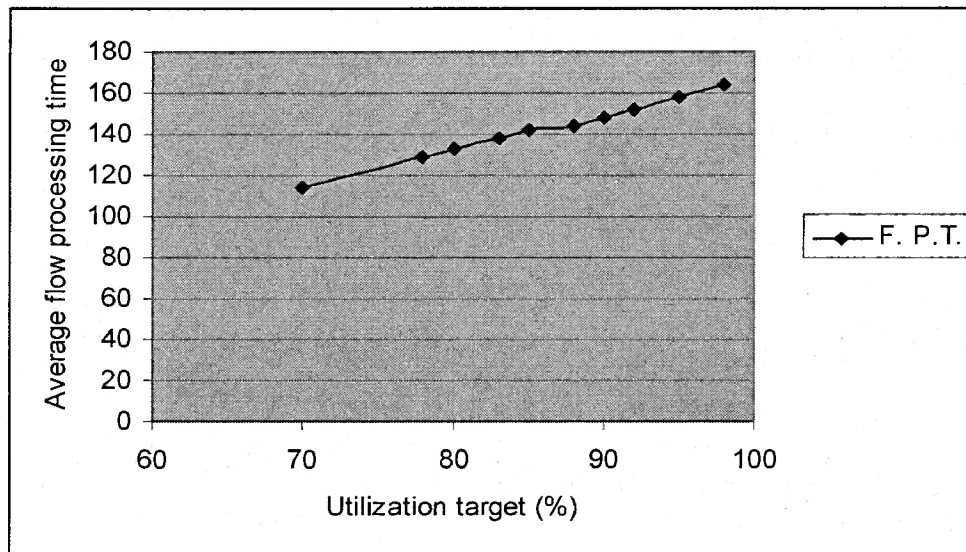


Figure 21 : Per-flow processing time with respect to link utilization

8.3.2 The probability that the needed resources will exceed the available resources

Prob.	F. P.T.
20	190
15	184
14	174
12.5	172
11	168
10	164
9	162
8	205
5	212
1	209
AVE	185

Table 13 : Per-flow processing time with respect to PNREAR

Table 15 and figure 22 show that increasing probability that the needed resource will exceed the available resources will decrease per-flow processing time. That is because this approach will be more conservative about admitting more calls.

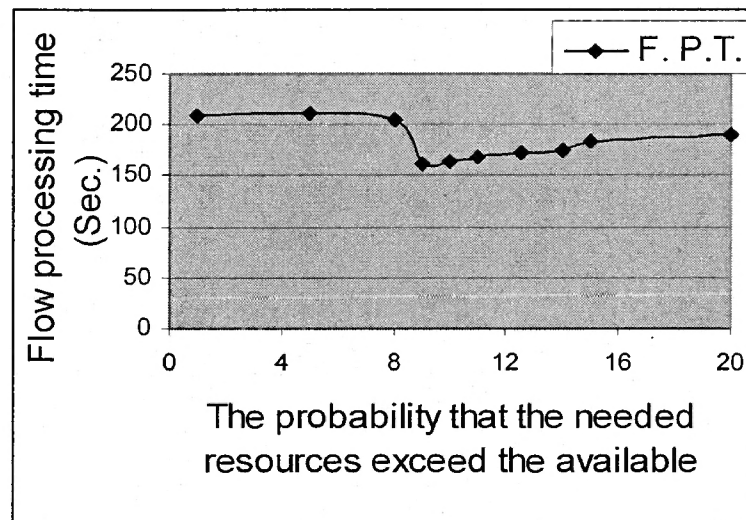


Figure 22: Flow processing time with respect to PNREAR

8.3.3 Per-flow processing time for the proposed approach

Table 16 and Figure 23 show that per-flow processing time decreases with respect to periodical estimation. In other words, if we make resource estimation every second, the average per-flow processing time is 205 sec. but if we do resource estimation every 10 seconds, the average per-flow processing time is 4.5 sec. this improvement is because of avoiding excessive resource estimation and because of the adaptation gains of the feedback mechanism.

Time	F. proc T.
1	205
2	191
3	156
4	85
5	25
6	20
7	9
8	4.5
9	4.5
10	4.5

Table 14 : per-flow processing time for the proposed approach with respect to time intervals

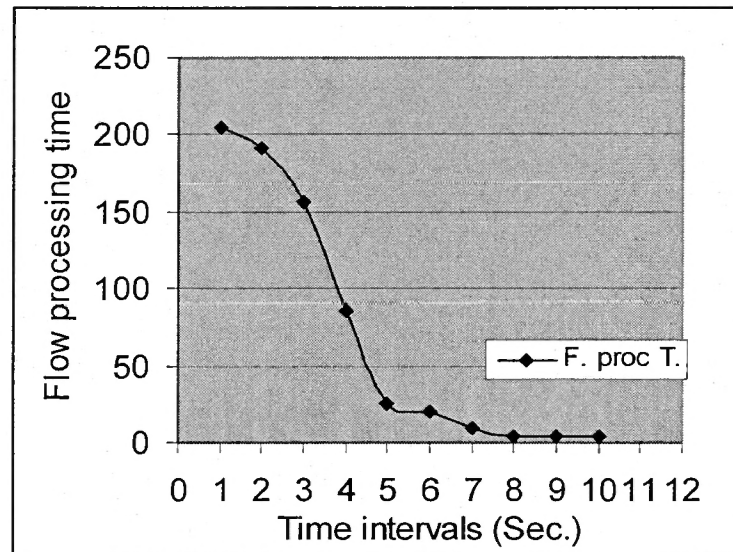


Figure 23: per-flow processing time for the proposed approach with respect to time intervals

8.3.4 Comparison

The proposed approach provides better QoS guarantee because of the deficiency of the other approaches and the feedback gains which we referred to previously. Taking the utilization target as a load-control parameter for estimation is a very strict and static approach. In addition, making estimations based on the probability that the required bandwidth will exceed the available resources is not a reliable approach. In addition, in our approach, we do periodical estimations but the other approaches perform per-flow estimations. Moreover, our approach has adaptive gains which come for the feedback mechanism.

	Call success rate (%)	Call rejection rate (%)	Flow processing time (%)
Util. T. b. IntServ	84	100	76
PNREAR B. IntServ	50	60	100
Renegotiation B. IntServ	100	57	37

Table 17: Comparing the percentage of Call admission, call rejection, and per-flow processing time for the different approaches

Table 17 and Figure 24 show the comparison among all approaches. The proposed approach provides the highest call admission ration and provides the lowest call rejection ration. In addition, it provides the best per-flow processing time. The proposed approach provides call success ratio 16 % higher than Utilization target based approach and 50% higher than PNREAR B. IntServ. In addition, it provides call rejection ratio 43 % less than utilization-target based approach and 40% less than PNREAR B. IntServ. The per-flow processing time for the proposed approach is 63% lower than PNREAR B. IntServ approach and 39 % lower than utilization target based approach.

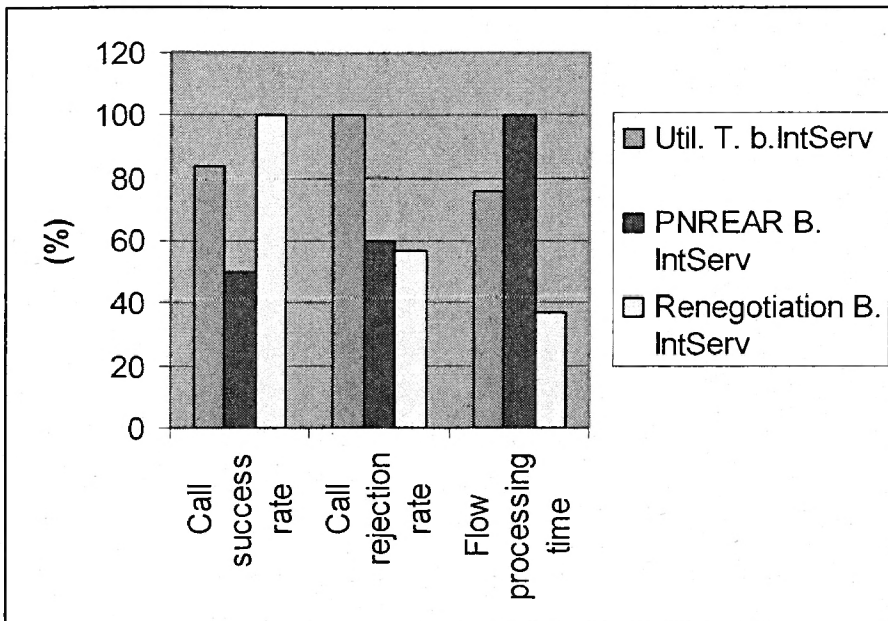


Figure 24: Comparing call admission ratio, call rejection ratio, and per-flow processing time for the different approaches

Chapter 9: Conclusion and future work

In this thesis, we proposed: (i) A new feedback mechanism based on call rejection notification, (ii) A new QoS parameter which is rejection notification. In addition, we considered not only the client but also the service provider by three things: a) Comparing overhead in different approaches with the overhead in the new proposed approach, b) Comparing how many calls have been handled and how many calls have been rejected in every approach, c) Calculating the utilization target and comparing it among the three approaches. The best approach is the one which does not only provide the best QoS but also which process the most number of flows and has the lowest call rejection rate.

The proposed feedback mechanism improves IntServ performance, achieves more fairness, decreases control overhead, and improves the QoS parameters. Our work shows that using a control parameter other than time or periodical resources' estimation gives non scalable QoS and a lot of control overhead. This is because of the per-flow excessive resource management.

Multimedia applications need guaranteed QoS and must have dynamic communication with the available resources so that applications can adapt to the available resources. In addition, they need an end-to-end controlled QoS support. The work shows that using a control parameter other than time or on-demand reaction from the resource estimator gives nonscalable QoS and a lot of overhead. This is because of per-flow and excessive resource reservation. This work has been done for single source and single destination. A good extension for this work is multi sources and multi destination network which is close to the reality. Another good potential extension for this work is examining the effect of

mobility on the performance of this approach. Another potential work in the future is doing the QoS dynamic by involving higher layers instead of doing it arbitrary. Service load control methods to provide specific QoS parameters still need much work in the future.

References:

- [1] Cristina Aurrecochea, Andrew T. Campbell, Linda Hauw, "A survey of QoS architectures, AC Multimedia System." J.-Special issue on QoS Architecture, May 1998.
- [2] Dan Chalmers and Morris Sloman, "A Survey of Quality of Service in Mobile Computing Environments," IEEE Communications Surveys, 2000.
- [3] J.C. Francis and M. Abu El-Ata, Benchmarking Mobile Networking QoS, Proc. Of the 36th Hawaii International Conference on System Sciences-2003.
- [4] Dong-Hoon Yi and Jong Won kim, Dynamic Resource Management Technique with Advance Reservation over QoS-provisioned Networks,
- [5] Yosuke Horibe and Yongbing zhang, "Adaptive QoS-Guaranteed Channel Reservation in Multimedia wireless networks." Int. Conf. communications, Circuits, and systems, 2002 CICCAS2002.
- [6] Ashish Desai, "An Adaptive QoS Mechanism for Multimedia Applications in Hetrogeneos Environments," Thesis from school of New Brunswick, 2001.
- [7] Bobby Vandalore, and el al, "AQuaFWiN: Adaptive QoS Framework for Multimedia in Wireless Networks and its Comparison with other QoS Frameworks," in Proc. 24th IEEE Conference on Local Computer Networking, 1998.
- [8] The COMET Group:<http://comet.ctr.columbia.edu/>
- [9] The RDRN Group: <http://www.ittc.ku.edu/Projects/RDRN/index.html>
- [10] E. Amir, S. McCanne, M. Vetterli, "A Layered DCT Coder for Internet Video," Proc. ICIP'96, Lausanne,Switzerland, Sep 1996.

- [11] I. Delgrossi, C. Halstrick, D. Hehmann, R. G. Herrtwich, O. Krone, J. Sandvoss, C. Vogt, "Media scaling for Audiovisual Communication with the Heidelberg Transport System," Proc. ACM Multimedia, pp 99-104, Jun 1993.
- [12] N. Yeadon, F. Garcia, D. Hutchison, D. Shepherd, "Filters :QoS Support Mechanisms for Multi-peer Communications," IEEE Journal of Selected Areas in Communication, vol. 14, no. 7, pp 1245-1262, Sep 1996.
- [13] Matheos Ioannis Kazantzidis, Adaptive Multimedia in Wireless IP Networks. Doctor of philosophy Dissertation, University of California Los Angeles.
- [14] Saleem N. Bhatti and G. Knight. "Enabling QoS adaptation decisions for Internet applications". Journal of Computer Networks and ISDN Systems. 4Q 1998.
- [15] Saleem N. Bhatti and Graham Knight. "QoS Assurance vs. Dynamic Adaptability for Applications." International Workshop on Network and Operating Systems Support for Digital Audio and Video. Proceedings NOSSDAV'98. July 1998.
- [16] Petia Todorova, and et al, Quality-of-Service-Oriented Media Access Control for Advanced Mobile Multimedia Satellite Systems.
- [17] C. Oliveira, J.B. Kim, and T. Suda. "An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks." IEEE J. Selected Areas Commun., 16(6):858-873, August 1998.
- [18] G.S. Kuo, P.C. Ko, and M.L. Kuo. "A probabilistic resource estimation and semi-reservation scheme for flow-oriented multimedia wireless networks." IEEE Commun. Mag., pp. 135-141, February 2001.

- [19] "Adaptive QoS-guaranteed channel reservation in multimedia wireless networks" (with Y. Horibe), Proc. Int. Conf. Communications, Circuits, and Systems 2002 (ICCCAS 2002), Chengdu, China, pp. 404-408 (Jun. 2002).
- [20] Tsu-Wei Chen, Paul Krzyzanowski, Michael R. Lyu, Cormac Sreenan and John Trotter, "A VC-Based API for Renegotiable QoS in Wireless ATM Networks," In Proceedings of ICUPC '97.
- [21] Mark A. Woodings, "A router agent capacity assessment in packet video," MS thesis, Texas A&M University, College Station, TX, Aug. 1997.
- [22] A. Bruce McDonald and Taieb Znati. "A Mobility-Based Framework for Adaptive Clustering in Wireless Ad-Hoc Networks." In IEEE Journal on Selected Areas in Communication, Vol. 17, No. 8, August 1999.
- [23] Balachander Krishnamurthy and Jia Wang, "On Network-Aware Clustering of Web Clients," Proceedings of ACM SIGCOMM'2000.
- [24] <http://www.lk.cs.ucla.edu/wamis.html>.
- [25] <http://www.faqs.org/rfcs/rfc3290.html>, Network Working Group, An Informal Management Model for Diffserv Routers, Y. Bernet, D. Grossman, S. Blake, A. Smith, May 2002.
- [26] <http://www.ietf.org/rfc/rfc3260.txt>, Network Working Group, New Terminology and Clarifications for Diffserv, April 2002.
- [27] Y. Bernet et. Al., "A Framework for Differentiated Services," IETF working draft <draft-ietf-diffserv-framework-02.txt>, Feb. 1999.

- [28] S. Blacke et al, "An architecture for differentiated services," IETF FRC 2475, Dec. 1998.
- [29] Prasanna Chaporkar, Saswati Sarkar "Providing Stochastic Delay Guarantees Through Channel Characteristics Based Resource Reservation in Wireless Network," Proceedings of Wireless Workshop on Mobile Multimedia, WoWMoM 2002, Atlanta, September 2002.
- [30] Leandros Tassiulas, Saswati Sarkar "Maxmin Fair Scheduling in Wireless Networks," Proceedings of INFOCOM 2002, pp. 763-772.
- [31] Distributed Fair Scheduling in a Wireless LAN. H. Vaidya, P. Bahl and S. Gupta ACM Mobicom'2000, Boston, Aug. 2000.
- [32] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. "Distributed Multi-Hop Scheduling with Delay and Throughput Constraints," ACM MOBICOM'2001, Rome, Italy, July 2001.
- [33] S. Sun and W. A. Krzyman, "Call Admission policies and Capacity Analysis of a Multi-Service CDMA Personal Communication System with Continuous and Discontinuous Transmission, " IEEE Vehicular Technology Conference, 1998, Pp. 218-223.
- [34] Z.L. Zhang, Z. Duan, L. Gao, and Y. T. Hou. "Decoupling QoS control from core routers: A novel bandwidth broker architecture for scalable support of guaranteed services." In Proc. ACM SIGCOMM, Sweden, August 2000.

- [35] Ibrahim Khalil, Torsten Braun: "Implementation of a Bandwidth Broker for Dynamic End-to-End Capacity Reservation over Multiple Diffser Domains." MMNS 2001: 160-174.
- [36] R. Braden et al, "Integrated services in the Internet architecture," IETF RFC 1633, 1994.
- [37] William Su and Mario Gerla. "Bandwidth Allocation Strategies for Wireless ATM Networks using Predictive Reservation," IEEE Globecom '98.
- [38] Mahmoud Nagshineh and Anthony S. Acampora, "QoS Provisioning in Micro-Cellular Networks Supporting Multimedia Traffic," INFOCOM'95, IEEE, p.107-84, April 1995.
- [39] Carlos Oliveira, Jaime Bae Kim, and Tatsuya Suda, "Quality-of-Service Guarantee in High-Speed Multimedia Wireless Networks," 1995 IEEE International Conference on Communications, IEEE, June 1996, p. 728-34.
- [40] A. Ramanathan and M. Parashar, "Active Resource Management for the Differentiated Services Environment," in Proc. of the international Conference on Internet Computing (IC'2001), June 2001.
- [41] Crawley, E., Berger, L., Berson, S., Baker, F., Borden, M., and J. Krawczyk, "A Framework for Integrated Services and RSVP over ATM", RFC 2382, August 1998.
- [42] Natalie Giroux and Sundhakar Ganti, Quality of Service in ATM networks, Prentice Hall, 1999 Pp 63-83.

- [43] Debasis Mitra and Martin Reiman and Jie Wang .Robust Dynamic Admission Control for unified cell and call QoS in statistical multiplexers, D. Mitra, M. I. Reiman and J. Wang, IEEE J. Sel. Areas in Commun., 16:5 (1998), pp. 692-707.
- [44] Dynamic Call Admission Control of an ATM Multiplexer with On/Off Sources, M. I. Reiman, J. Wang and D. Mitra, Proceedings of the 34th Conference on Decision and Control, IEEE, 1995, pp. 1382-1388.
- [45] G.S. Kuo and po-chang, "A Probabilistic Resource Estimation and Semi-Reservation Scheme for Flow-Oriented Multimedia Wireless Networks, "IEEE Communication magazine, Feb. 2001.
- [46] A. I. Elwalid AND D. Mitra. Fluid Models for the Analysis and Design of Statistical Multiplexing with Loss Priorities on Multiple Classes of Bursty Traffic. IEEE Trans. Communications, Vol.42, No.11, 2989-3002, 1992.
- [47] Sugih Jamin and Scott J. Shenker et al, "Comparison of Measurement-BASED Admission Control Algorithms for Controlled-Load Services"
< <http://citeseer.nj.nec.com/jamin97comparison.html>.
- [48] G. Smyth, et al, "Wireless: Growth Engine for Advanced Global Telecommunications, "in 1995 IEEE MTT-S Symposium on Technologies for Wireless Applications Digest, pp. 7-11, Feb. 1995.
- [49] E. Geraniotis, Y.W. Chang and W.B. Yang, "Multimedia Integration in CDMA Networks," IEEE third Interntional Symposium on Spread Spectrum Techniques and Applications, Vol 1, pp.88-97.

- [50] Sirisha R. Medidi, "A Uniform Policy for Handoff in Mobile Wireless ATM"
"http://athena.csie.ndhu.edu.tw/1003-1.ppt.
- [51] M. Oliver, J. Borràs, Performance Evaluation of Variable Reservation Policies for Hand-off Prioritization in Mobile Networks, Proceedings of INFOCOM'99, New York.
- [52] C. J. Chang, T. T. Su, "A channel borrowing Scheme in a Cellular Radio System with guard channels and Finite Queues" in ICC'96, PP. 1168-1172.
- [53] O. T. Yu and V. C. Leung, "Adaptive resource allocation for prioritized call admission over an ATM-BASED WIRELESS pcn," IEEE J. Selected areas COMMUN., vol.15, pp. 1208-1225, Sept. 1997.
- [54] Fei Yu, Victor Leung. Mobility-based predictive call admission control and bandwidth reservation in wireless cellular networks , Computer Networks: The International Journal of Computer and Telecommunications. Volume 38, Issue 6 April 2002. Pages: 577 – 589.
- [55] Vergados D.D., Anagnostopoulos, et al, Call Admission Control with Adaptive allocation of Resources in Wirelss ATM Networks, IEEE MELECON 2002.
- [56] Vergados D.D., Anagnostopoulos C.E., Anagnostopoulos et al, Call Admission Control with Adaptive Allocation of Resources in Wireless ATM Networks, IEEE MELECOM 2002.
- [57] Mohmoud Nagshineh and Anthony S. Acampora, "QoS Provisioning in Micro-Cellular Networks Supporting Multimedia Traffic", INFOCOM'95, IEEE, p1075-84, April, 1995.

- [58] V. G. Kulkarni and N. Gautam, "Admission Control of multi-class traffic with service priorities in high-speed networks,"
- [59] Sirisha R. Medidi, "A Uniform Policy for Handoff in Mobile Wireless ATM Networks," In proceedings of IEEE International Performance Conference on Computing and Communication (IPCCC), p. 201-207, Phoenix, AZ, February 2000.
- [60] Sirisha R. Medidi and Forouzan Golshani, "Design Issues for efficient handoff in Wireless ATM Networks," To appear in the Journal of Annual Reviews of Communication, vol. 55, 2003.
- [61] J. M. McManus and K. W. Ross. "Video on demand over ATM networks: constant-rate transmission and transport." In Proc. IEEE INFOCOM, San Fransisco, CA, March 1996.
- [62] A. R. Reibman and A. W. Berger. "Traffic descriptors for VBR video teleconferencing over ATM networks." IEEE/ACM Transportation on Networking, 3(3):329-339, June 1995.
- [63] Yousif Iraqi, Raouf Boputaba and Alberto Leon-Garcia. "QoS control in Wireless ATM," Mobile Networks and Application 5 (2000) 137-145.
- [64] John Sikora and Ben Teitelbaum, Differentiated Services for Internet2, <http://qos.internet2.edu/may98Workshop/html/diffserv.html>, January 2004.
- [65] Rolf Neugebauer and Derek McAuley, "Congestion Prices as Feedback Signals: An Approach to QoS Management," In Proc. of the 9th ACM SIGOPS European Workshop, Kolding, Denmark, September 2000.

- [66] B. Li and K. Nahrstedt. "A Control Theoretical Model for Quality of Service Adaptations." Proceedings of Sixth International Workshop on Quality of Service, 1998.
- [67] Baochun Li, Dongyan Xu, Klara Nahrstedt. "Optimal State Prediction for Feedback-Based QoS Adaptations," in Proceedings of Seventh IEEE International Workshop on Quality of Service (IWQoS 99), pp. 37-46, London, UK, May 31 - June 4, 1999.
- [68] Jeong Geun Kim, Marwan M. Krunz. "Bandwidth allocation in wireless networks with guaranteed packet-loss performance." IEEE/ACM Transactions on Networking (TON) Volume 8 , Issue 3 (June 2000) Pages: 337-349.
- [69] Ingo Busse, Bernd Deffner, Henning Schulzrinne, "Dynamic QoS Control of Multimedia Applications based on RTP," Computer Communications, vol. 19, no. 1, pp. 49-58, January 1996.
- [70] J.-C. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the internet," in SIGCOMM Symposium on Communications Architectures and Protocols, (London, England), pp. 58--67, ACM, Aug. 1994.
- [71] H. Kanakia, P. Mishra, and A. Reibman, "An adaptive congestion control scheme for real-time packet video transport," in SIGCOMM Symposium on Communications Architectures and Protocols, (San Francisco, California), pp. 20--31, ACM/IEEE, Sept. 1993.
- [72] S. Rampal et al., "Dynamic Resource Allocation Based on Measured QoS," Technical Report TR 96-2, Center for Advanced Computing and Commun., North Carolina State University, Jan. 1996.

- [73] I. Hsu and J. Walrand, "Dynamic Bandwidth Allocation for ATM Switches," *J. Applied Probability*, vol. 33, no. 3, 1996, pp. 758–71.
- [74] P. Agrawal et al., "SWAN: A Mobile Multimedia Wireless Network," in *IEEE Personal Communications*, Apr. 1996. PP. 18-33.
- [75] G. Kesidis, "Bandwidth Adjustments Using Online Packet-level Measurements," *SPIE Conf. Performance and Control of Network Systems*, Boston, MA, Sept. 1999.
- [76] Peerapon Siripongwutikorn, Hewlett-Packard Laboratories; David Tipper, "A Survey of Adaptive Bandwidth Control Algorithms," University of Pittsburgh
- [77] http://www.cisco.com/warp/public/cc/so/neso/vvda/ipatm/ipne_wp.htm
- [78] The COMET Group: <http://comet.ctr.columbia.edu/>
- [79] The RDRN Group: <http://www.ittc.ku.edu/Projects/RDRN/index.html>
- [80] [113] S. Floyd. "Comments on Measurement-based Admission Control for Controlled-Load Services". Submitted to *Computer Communication Review*, 1996.
- [81] S. Floyd. "Comments on Measurement-based Admission Control for Controlled-Load Services". Submitted to *Computer Communication Review*, 1996.