

**ACTIVE RESOURCE MANAGEMENT FOR THE DIFFERENTIATED
SERVICES ENVIRONMENT**

by

ANANTHANARAYANAN RAMANATHAN

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

And approved by

Professor Manish Parashar

New Brunswick, New Jersey

May 2002

ABSTRACT OF THE THESIS

Active Resource Management for the Differentiated Services Environment

by ANANTHANARAYANAN RAMANATHAN

Thesis Director:

Professor Manish Parashar

In the current Internet architecture, a large percentage of the traffic is either multimedia related or a form of real time data that is critical to an application. Typical applications include Voice over IP (VoIP) and video conferencing. These time-critical data require some level of Quality of Service (QoS) guarantee. QoS is the classification of packets for the purpose of treating certain classes or flows of packets in a particular way as compared to the other packets. The Internet Protocol (IP), however, is based on best effort and lacks the capability to provide such QoS guarantees. Various solutions have been proposed to address this problem by guaranteeing applications their required resources. These include integrated services (e.g. RSVP), differentiated services and Multi Protocol Label Switching (MPLS).

The Differentiated Services (DiffServ) network architecture attempts to provide these QoS guarantees in the most scalable and least complex manner. A DiffServ domain defines two levels of service provisioning: the standard best effort service, which is similar to IP, and the premium services where the clients' requests for service

guarantees are met. In the DiffServ architecture, the Bandwidth Broker (BB) manages a domain's resources using service policies defined based on the client's requirements. The BB reserves the bandwidth requested by a client for a price. However, this reservation is made without any understanding of the nature of the information that will be transmitted. Although such a reservation provides a better sense of resource allocation than that provided by the DiffServ domain on its own, the result is still a static provisioning of resources, and can lead to wasted bandwidth.

This thesis presents the active resource management (ARM) approach that actively manages the resources in a DiffServ domain by dynamically reallocating resources based on the current requirements of applications and the state of the network. The ARM approach is motivated by the observation that the actual traffic generated by a client rarely approaches the peak rate bandwidth that has been reserved for the client. Consequently, when the client's traffic drops below the reserved rate, a portion of the unused bandwidth can be returned to a pool of available bandwidth. ARM is implemented and evaluated using the network simulator (NS-2) toolkit. Our experiments demonstrate that by actively over provisioning and dynamically reallocating available resources, ARM can effectively increase the overall utilization of the available bandwidth, and support increased number of clients while honoring QoS guarantees.

Acknowledgements

I thank Professor Manish Parashar for providing me an environment to conduct excellent research and for being a patient thesis advisor to me. He has built a great research lab, TASSL, and I am grateful that he has allowed me to be a part of it.

I thank Professor Dan Reininger for providing me with a lot of initial guidance in developing the basic idea behind this thesis. He also taught me a great deal about communication networks.

I also thank everyone in the TASSL lab for making the past three years a wonderful experience.

I really appreciate all the support and encouragement I received from my seniors especially from Samian Kaur and Rajeev Muralidhar. I would also like to thank my friend Firdaus for helping me format, edit, and proofread this thesis.

My stay in Rutgers has been most enjoyable thanks to my roommates Aaresh, Ashutosh and Som in Buell 302, and Anirudha, Akshay, and the rest in Riverside Apartments 8R and 10R.

Last and the Most, I thank my family for their patience and understanding, love and constant encouragement throughout my life.

Table of Contents

ABSTRACT OF THE THESIS	ii
Acknowledgements.....	iv
List of Illustrations	vii
Glossary of Terms.....	viii
Chapter 1	1
Introduction.....	1
1.1 Objective	1
1.2 Problem Statement	1
1.2.1 Quality of Service.....	1
1.2.2 Differentiated Services.....	2
1.2.3 Bandwidth Broker.....	2
1.3 Overview of the Active Resource Management algorithm	2
1.4 Contributions of the Thesis	3
1.5 Thesis Overview.....	3
Chapter 2.....	4
Background and Related Work	4
2.1 Background	4
2.1.1 Differentiated Services	4
2.1.2 Bandwidth Broker.....	7
2.2 Related Work.....	9
2.2.1 Resource Management	10
2.2.2 Resource Management in DiffServ	10
2.2.3 Active Resource Management in DiffServ	11
2.2.4 A Comparison of Active Resource Management Mechanisms	13

Chapter 3	14
Active Resource Management (ARM)	14
3.1 Architectural Framework.....	14
3.2 Illustrative Examples.....	16
3.3 Implementation Using Network Simulator 2	18
Chapter 4.....	21
Experimentation and Evaluation.....	21
4.1 Experiments.....	23
4.1.1 Experiment 1: Exact Allocation of Resources	23
4.1.2 Experiment 2: Over Allocation of Resources	26
4.1.3 Experiment 3: Over Allocation for an Extended Period of Simulation	29
Chapter 5.....	31
Summary and Future Work.....	31
5.1 Summary.....	31
5.2 Future Work.....	31
References.....	33

List of Illustrations

FIGURE 1 THE DIFFSERV BOUNDARY AND INTERIOR ELEMENTS	5
FIGURE 2 A GENERIC MODEL OF A BANDWIDTH BROKER IN A DS DOMAIN.....	6
FIGURE 3 FUNCTIONAL DECOMPOSITION OF THE BANDWIDTH BROKER.....	8
FIGURE 4 HETEROGENEOUS NETWORK SHOWING DIFFERENT SCENARIOS	17
FIGURE 5 MODULAR BREAKUP OF THE BB AND ITS INTERACTIONS IN A TRAFFIC FLOW	19
FIGURE 6 OUR TEST NETWORK	21
TABLE 1 A COMPARISON OF DIFFERENT ACTIVE RESOURCE MANAGEMENT SCHEMES	13
TABLE 2 LEGEND FOR THE TEST NETWORK.....	22
TABLE 3 INITIAL POLICY REQUEST.....	22
TABLE 4 POLICY TABLE FOR EXPERIMENT 1	23
TABLE 5 PACKET STATISTICS FOR EXPERIMENT 1	25
TABLE 6 POLICY TABLE FOR EXPERIMENT 2.....	26
TABLE 7 PACKET STATISTICS FOR EXPERIMENT 2	27
GRAPH 1 A PLOT OF BANDWIDTH USED VS TIME FOR EXPERIMENT 1	24
GRAPH 2 A PLOT OF THE PACKET STATISTICS FOR EXPERIMENT 1.....	25
GRAPH 3 A PLOT OF BANDWIDTH USED VS TIME FOR EXPERIMENT 2	27
GRAPH 4 A PLOT OF THE PACKET STATISTICS FOR EXPERIMENT 2.....	28
GRAPH 5 A PLOT OF BANDWIDTH USED VS TIME FOR EXPERIMENT 3	29
GRAPH 6 A PLOT OF THE PACKET STATISTICS FOR EXPERIMENT 3.....	30

Glossary of Terms

AF	Assured Forwarding: A PHB group consisting of four PHB classes. AF PHB provides quality differentiation related to various quality aspects.
BA	Behavior Aggregate: A collection of packets with the same DS codepoint crossing a link in the particular direction.
BE	Best Effort Services: A service model without explicit quality guarantees.
DiffServ	Differentiated Services: A quality of service model.
DS	Differentiated Services
DSCP	Differentiated Services Code Point: The first six bits in the TOS header of an IP packet.
EF	Expedited Forwarding: Premium service class as defined in the DS architecture.
IntServ	Integrated Services: Another quality of service model.
Marking	A process of setting bits in a packet header in order to have an effect on the treatment of the packet.
Metering	A measuring process of the temporal properties of a traffic stream.

MFC	Multi-Field Classifier: It selects the packets based on the content of some combination of source address, destination address, DS field, protocol ID, source port and destination port.
MPLS	Multi-Protocol Label Switching: Another quality of service model.
PHB	Per Hop Behavior: The externally observable forwarding behavior applied at a DS complaint node to a DS behavior aggregate.
QoS	Quality of Service: A set of attributes that can be used to define the network's capabilities to meet the requirements of users and applications.
RSVP	Resource Reservation Protocol: A protocol implemented by the IntServ model to help provide quality of service.
SLA	Service Level Agreement: A service contract between a customer and the service provider that specifies the forwarding service a customer should receive.
Shaping	A mechanism that delays packets of a traffic stream to achieve better network efficiency or to ensure conformance with a traffic profile.
TOS	Type of Service: An 8 bit field in the IPv4 packet header designed to indicate the preferred treatment of the packet.

Chapter 1

Introduction

1.1 Objective

There is a need for service guarantees in a world of best effort services. IP on its own does not provide any QoS. Differentiated Services over IP along with bandwidth broker provide a level service guarantees with static provisioning of resources. Our objective is to optimize resource utilization within a network by dynamically reallocating the unused bandwidth allotted to a client and if possible increase the number of clients receiving guaranteed service.

1.2 Problem Statement

1.2.1 Quality of Service

Quality of Service has been defined as “ The network’s capability to provide a non-default service to a subset of the aggregate traffic”[2]. In other words, QoS refers to the classification of packets so that classes or flows of packets can be treated in a specific way when compared to other packets. The Internet today hosts various types of applications with heterogeneous requirements for e.g. Voice over IP and packetized video present multimedia data in real time; while low-bandwidth text based business messaging systems requiring “high-priority” due to their mission critical information. In both cases the network must handle application packets in a special way so as to provide service guarantees in terms of bandwidth, jitter, latency and other data transfer parameters. But the Internet and, more generally, IP networks offer no easy way to identify such packets or give them special handling [2]. Various solutions have been proposed to address this problem and guarantee applications their required resources.

These include Integrated Services [18] (e.g. Resource reSerVation Protocol (RSVP) [20]), Differentiated Services and Multi Protocol Label Switching (MPLS).

1.2.2 Differentiated Services

The Differentiated Services (DiffServ) network architecture for QoS management attempts to provide QoS guarantees in the most scalable and least complex manner. It does this by defining different classes of service provisioning such as the standard best effort service (BE), the predictive service or Assured Forwarding (AF) and guaranteed service or Expedited Forwarding (EF) [3]. In order to provide guaranteed service, the DiffServ allocates its resources (i.e. bandwidth) to the client according to the clients' needs. This allocation is static, wherein the amount of bandwidth allocated belongs to that particular client whether or not the client uses the bandwidth.

1.2.3 Bandwidth Broker

The bandwidth broker is a middleware entity in a DiffServ environment and it provides a centralized and intelligent reservation mechanism [16]. BB's can be configured with the clients' requirements, defined in terms of service policies, to keep track of current allocation of marked traffic and to interpret new requests and re-mark the traffic in light of the existing service policies and current allocation. In short, the BB architecture makes it possible to maintain state on an administrative domain basis, rather than at every router, furthermore the service definitions of Premium and Assured service make it possible to confine the per flow state to just the leaf routers and significantly reduce the complexities of management at the core routers. Note that, although there is a better understanding of allocation of resources within the DiffServ domain, the result is still a

static provisioning of resources that can lead to wasted bandwidth and starved applications.

1.3 Overview of the Active Resource Management Algorithm

It has been observed that the actual traffic generated by a client rarely approaches the peak rate bandwidth that the client reserves for the duration of the flow. The BB and the DiffServ architecture allocate the peak request for bandwidth to the client and that bandwidth cannot be used by any other flow for the duration it has been allotted. The ARM approach attempts to actively manage the resources of the DiffServ domain by dynamically reallocating the resources based on the current requirements of the application and the state of the network. This is achieved by keeping a track of the traffic flow of each client, and dynamically reclaiming a part of the unused bandwidth and reallocating it to other flows, thus increasing the number of flows that can be allotted resources, and improving the resource utilization within a DiffServ domain.

ARM is implemented and evaluated using the network simulator (NS-2) [5] toolkit. Our experiments demonstrate that by actively over provisioning and dynamically reallocating available resources, ARM can effectively increase the overall utilization of the available bandwidth by about 25%, depending on the clients' application, and supports increased numbers of clients while honoring QoS guarantees.

1.4 Contributions of this thesis

This thesis aims at providing a mechanism to actively manage the existing resources within a DiffServ environment. The major contributions of this thesis are:

1. We have designed the Active Resource Management algorithm which helps the BB manage the network resources actively to provide:
 - a. Better resource utilization of about 25%, and

- b. More clients with the requisite service guarantees.
- 2. We have designed a bandwidth broker entity, in accordance with the basic requirements of the Qbone signaling design team, for the NS-2 toolkit.

1.5 Thesis Overview

The rest of this thesis is organized as follows. Section 2 presents the background material and outlines the related work. Section 3 describes the ARM approach and the implementation issues, section 4 explains the experiments and gives a thorough evaluation using the NS-2 network simulator, and presents experimental results. Section 5 presents our conclusions.

Chapter 2

Background and Related Work

2.1 Background

Existing models for providing Quality of Service are based on service differentiation. These models can be classified as reservation protocols, label switching and relative priority marking [7]. The reservation protocol model, such as RSVP, relies upon traditional datagram forwarding in the default case, and uses an exchange of signaling messages to establish packet classification and forwarding state on each node along the data transfer path. This requires the maintenance of state at each hop along the path for the duration of the transfer, reducing its scalability. The label-switching model includes MPLS and ATM. In this model, path forwarding state and traffic management is established for traffic streams on each hop along the network path permitting finer granularity resource allocation to traffic streams. This improved granularity comes at the cost of additional management and configuration required to establish and maintain the label switched paths. In addition, the amount of forwarding state maintained at each node scales in proportion to the number of edge nodes of the network in the best case (assuming multipoint-to-point label switched paths), and it scales in proportion with the square of the number of edge nodes in the worst case, assuming edge-edge label switched paths with provisioned resources are employed. We have accepted DiffServ as the least complex and scalable solution for providing QoS.

2.1.1 Differentiated Services

Differentiated Services is a set of technologies that are used to provide quality of service (QoS) in a world of best effort service provisions [2]. It provides a framework

and building block to enable deployment of scalable service discrimination in the Internet. To achieve scalability, the individual host-to-host microflows are aggregated into a single larger microflow and the aggregate flow receives special treatment. The DiffServ architecture [7] is based on a simple model where the traffic entering a network is classified and possibly conditioned at the boundaries of the network, and then assigned to different behavior aggregates, thus pushing all the complexities to the edge routers, leaving the core routers as simple as possible. DiffServ classifies individual microflows at the edge routers in the network, into one of the many classes and then applies a per-class service in the core of the network. This classification is performed at the network's ingress router, based on the service requested, and marked with a DiffServ Code Point (DSCP) in the ToS (Type of Service) field of Ipv4. The core routers that forward the packet, associated with a flow, examine the DSCP to determine how the packet should be treated. All packets marked with the same DSCP form a behavior aggregate

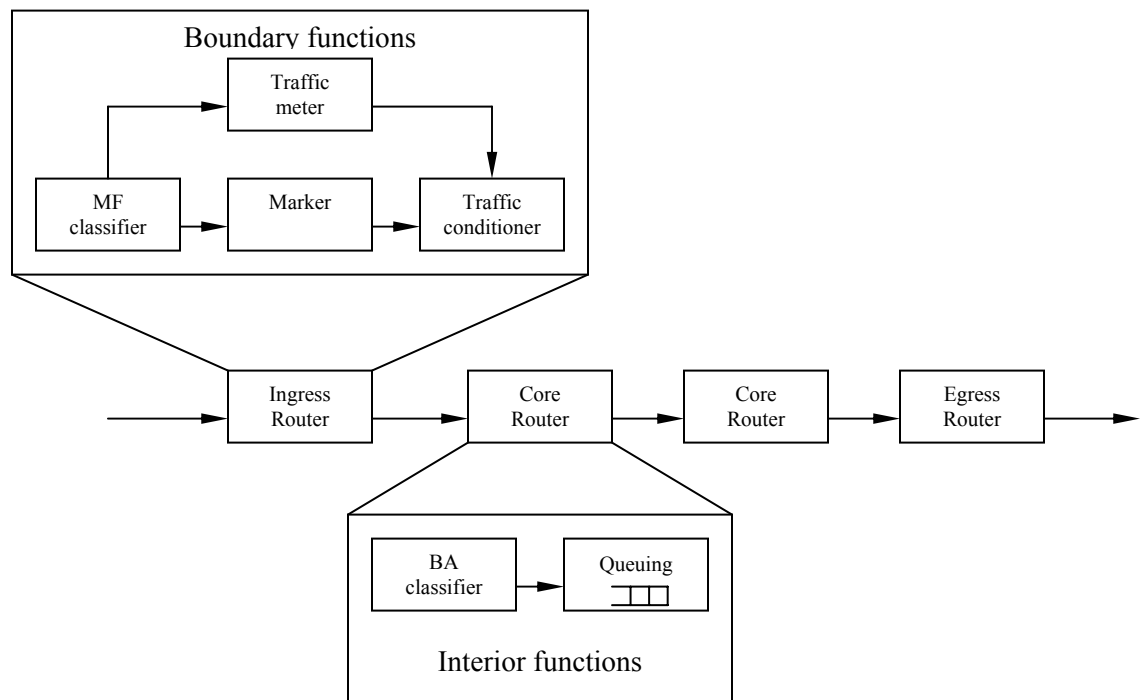


Figure 1: The DiffServ architecture.

(BA); and a Per Hop Behavior (PHB) is applied to each BA inside the network. The PHB defines the service the packet receives at each hop as it is forwarded through the network. In the figure below, the boundary router or the edge router is positioned at the edge of the DiffServ capable network. This router is responsible for packet classification, packet marking, metering and traffic conditioning. Interior nodes are core switches or routers that provide the PHB based on the DSCP bits. The core routers employ queue management techniques and scheduling mechanisms such as random early detection (RED) and weighted fair queuing (WFQ) to provide the PHB. There are two defined PHB's: expedited forwarding (EF), and assured forwarding (AF). EF PHB [34] is defined to support low loss, low delay, and low jitter. The AF PHB [8] defines four relative classes of service with each service supporting three levels of drop precedence.

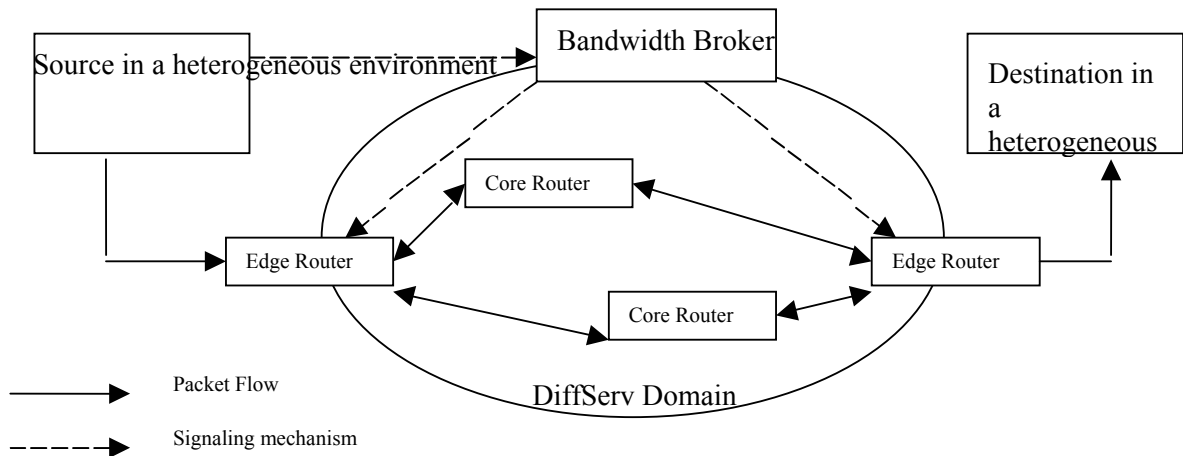


Figure 2: A generic model of a bandwidth broker in a DS domain.

DiffServ is being regarded as a reasonable solution to provide Quality of Service on the Internet. Research and testing of the DiffServ architecture is being conducted by TF-TANT [35] for a European environment, CSIRO and AARNET (Australian Academic

and Research Network) [36] for an Australian environment and by other universities such as the University of Kansas [38] and Massachusetts Institute of Technology [37]. Vendors such as Cisco, IBM, Nortel Networks, Lucent, Cabletron and Ericsson [36] provide DiffServ functionality in their routers, and Nortel Networks has evaluated DiffServ using NS-2 toolkit.

2.1.2 Bandwidth Broker

A bandwidth broker [16] manages network resources for IP QoS services that are supported within a network and used by customers of the network services. A BB may be considered a type of policy manager in that it performs a subset of policy management functionality such as access of users to network services. The SLA is a Service Level Agreement (SLA) is a service contract between a customer and a service provider that specifies the forwarding service a customer should receive. The Service Level Specification (SLS) is a translation of the SLA into appropriate information necessary for provisioning and allocating QoS resources within the network devices, in particular, at the edges of the domain on links connecting the domain to adjacent domains. The bandwidth broker requires both the SLA and the SLS to achieve a range of services accorded to the user. Based on the SLA the broker decides whether it can provide the requested allocation, and it configures the edge router accordingly to mark and classify the packets as decided in the SLS [7]. The BB is also responsible for managing inter-domain communication, with BBs in neighboring networks to coordinate SLSs across the domain boundaries.

The BB gathers and monitors the state of QoS resources within its domain and on the edges of the domain to and from adjacent domains. This information, together with the policy (from the policy rules database) is used for admission control decisions on QoS

service requests to the network. The network state information from the BB is used to verify that resources are currently available in the network to support a request. Across boundaries, the SLS may be on an aggregate basis, where aggregation is on all flows within the domain of a particular QoS service type (i.e. DiffServ codepoint). Within a domain, individual flows may be allocated resources based on the SLS by issuing Resource Allocation Requests (RARs). It is the responsibility of the BB to coordinate allocation and provisioning of the aggregate resources of the SLSs, into and out of its domain, with the resources requested by the RARs.

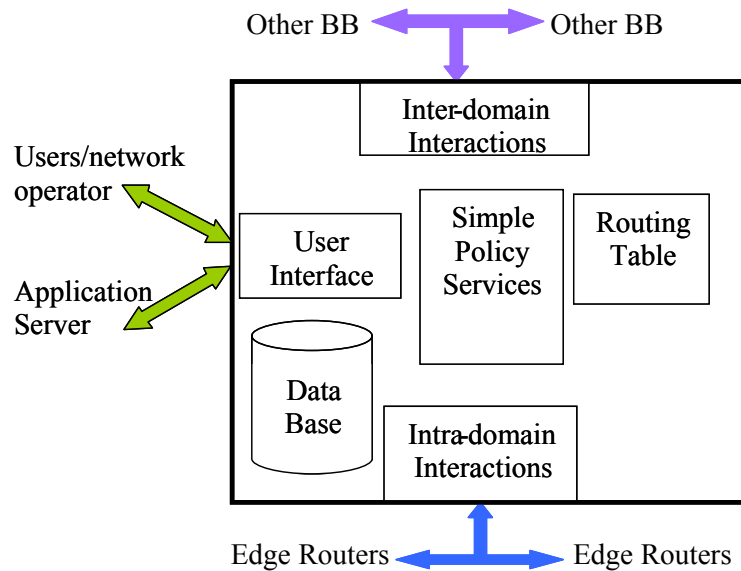


Figure 3: Functional decomposition of the bandwidth broker as defined by Qbone [14]
team

The bandwidth broker consists of a few basic components as shown in Figure 3. Their functions are defined as follows [14]:

1. User Interface: The user/application interface provides a means for the user to make resource requests directly, or to the network operator who will enter requests. The interface also receives messages from setup protocols.

2. Inter-domain Interactions: A method for enabling peer BBs' to make requests for resources and admission control decisions to enable flow of traffic.
3. Intra-domain Interactions: Providing a method for enabled BB to configure the edge routers within the domain to provide quality of service.
4. Routing Table: A routing table is maintained to access inter-domain routing information so that BB can determine the edge routers and the downstream routers before allocating their resources. Also additional routing paths can be maintained for different flows within the domain.
5. Database: A database is used to store information about all the BB's parameters. The different information stored within the repository are: SLAs, current reservations, configuration of routers, DSCP mapping, and policy information.

The Bandwidth Broker has been designed to add intelligence to the DiffServ, to help optimize the existing resources. The Internet2 Qbone Bandwidth Broker Advisory Council (BBAC) has defined a set of standards for BB, and work is being carried on at the Lawrence Berkeley National Labs (LBNL) [39], University of Kansas [38], Siemens, and various universities. There are two commercial products with BB capabilities made available by Orchestream [40] and Extremeware [41].

2.2 Related Work

QoS is the current buzzword on the Internet. Quality of Service can be achieved through efficient resource allocation by network architecture or by software programs that provide good resource management. Of the many network architectures suggested, such as IntServ (RSVP), MPLS and others, DiffServ provides the most scalable and least complex solution. Resource management is a relatively new field that is a popular research area as it is easier and necessary to manage existing resources for

heterogeneous networks, than it is to design and implement a new network architecture that provides QoS.

2.2.1 Resource management

Key works in resource management include PBNM, and GARA. Intel's Policy Based Network Management (PBNM) [25] technology provides the ability to define and distribute policies to manage heterogeneous networks. These policies control critical network resources such as bandwidth, security, and Web access. PBNM is scalable and offers multi-vendor support. The Globus group defines the Globus Architecture for Reservation and Allocation (GARA) [26] [27] that supports flow specific QoS specifications, immediate and advance reservations, and online monitoring and control of both individual resources and heterogeneous resource ensembles. The prototype GARA implementation builds on the differentiated services mechanisms to enable the coordinated management of two distinct flow types - foreground media flows and background bulk transfers, as well as the co-reservation of networks, CPU's and storage systems. Other works in resource management include resource allocation schemes for connections' tolerating statistical QoS guarantees in public wide area ATM networks using effective bandwidth [28].

2.2.2 Resource Management for the DiffServ

Many schemes have been proposed to help better manage the resources in the DiffServ architecture. The scheme proposed by Reichmeyer and Zhang, defines a two-tier resource management model [16] and introduces the concept of Bandwidth Broker as the resource manager for each domain and a BB-to-BB protocol equivalent to BGP in routing, for inter-domain resource management. The BB is one of the more popular and accepted resource management concepts for the DiffServ. The second scheme proposed

by Jun Ogawa and Yuji Nomura [15] is an extension to the first one, wherein it is shown that the BB is not sufficient to manage the resources due to the existence of multi-vendor routers running various routing protocols within the same administrative domain. Since DiffServ specifies only externally observable behaviors in the Forwarding Path, equipment vendors can use different mechanisms to implement these behaviors and if BBs were to directly control each router they would have to be aware of the details of each router's implementation. This level of detail complicates the design of a BB, especially in large heterogeneous domains where a number of different router designs coexist. Thus the second approach proposes the design of an edge router be equipped with *Virtual Configuration Manager* (VCM), where a higher-level description named *Virtual Configuration Description* (VCD) generated in BB is translated into the specific parameters for the Forwarding Path. As VCM succeeds to veil the details of the Forwarding Path implementation, BB can concentrate on the management of SLA without being aware of different implementations of routers. This is made possible by designing the VCM in Java and making it OS independent.

2.2.3 Active Resource Management for the DiffServ

Resource management on the Internet is essential and the introduction of DiffServ and Bandwidth broker to further satisfy these needs is appropriate. However these methods provide a static one-time management at the start of a flow, resulting in resource wastage. For current network traffic, which is dynamic in nature, we need to actively manage existing resources so as to optimize utilization, to reduce network congestion and provide QoS. One of the first approaches towards this goal was the Active Queue Memory Management concept [31] that actively manages a DiffServ's queues in order to provide better and more proactive response to network congestion to meet QoS goals.

The two mechanisms that support active queue management in large IP networks are: Random Early Detection (RED) (currently being deployed in large IP networks along with various extensions to RED such as Weighted RED (WRED)), and Explicit Congestion Notification (ECN) (initially an experimental addition to the IP architecture). Another proposed solution is the Resource Management in DiffServ (RMD) [18] framework for radio access networks (RAN). The current strategies for resource management do not meet the requirements for resource management within a RAN. It is a lightweight solution for the edge-to-edge dynamic resource management problem of DiffServ domains. The RMD framework follows a distributed admission control and resource management approach, which is different from the bandwidth broker designs for DiffServ. The RMD framework for the DiffServ is an open framework interoperable with other resource management mechanisms with wide scope of applicability in different DiffServ networks. It is a simple framework with good scaling properties and has low cost of implementation. Since it is designed for extension of DiffServ concepts, the RMD relies on DiffServ principles for QoS provisioning and preserves its scalability properties. The RMD framework enhances these concepts with new ones necessary to provide dynamic resource provisioning and admission control in DiffServ domains. In the RMD framework the problem of a complex reservation within a domain is separated from a simple reservation within a node. Accordingly there are two types of protocols defined within the RMD: Per Hop Reservation (PHR) and Per Domain Reservation (PDR). The PHR is a newly defined protocol, while the PDR could be one of the existing protocols such as RSVP [20], RSVP Aggregation, Simple

Network Management Protocol (SNMP) [42], and Common Open Policy Service (COPS) [43].

Approach presented in this thesis provides Active Resource Management to the DiffServ environment. It uses the bandwidth broker as the resource manager and proposes an improvement in the reservation mechanism by giving it run-time capabilities. Resource utilization is improved by keeping a track of the network characteristics per client flow and reusing the unused, already allocated, bandwidth without loss of service, thus preventing wastage of resources, and increasing the number of clients who can receive service.

2.2.4 A Comparison of Active Resource Management mechanism

	Active Queue Memory Management	RMD	ARM
Target Networks	Heterogeneous Networks	Radio Access Networks	Heterogeneous Networks
Resource Manager	None	None	Bandwidth Broker
Changes made to DiffServ	All the Router Queues	DiffServ's Per Hop Behavior Functionality	Bandwidth Broker
Contributions	Random Early Detection (RED) & Explicit Congestion Notification (ECN)	Per Hop Reservation (PHR) & Per Domain Reservation (PDR)	Active Resource Management (ARM) Algorithm
Results	Manages congestion and end-to-end delay and supports delivery of DiffServ classes.	Improves resource utilization and provides better resource management capabilities to RANs	Improves resource utilization, reduces resource wastage and increases the number of clients.

Table 1: A Comparison of Different Active Resource Management Schemes

Chapter 3

Active Resource Management (ARM)

3.1 Architectural Framework:

As described above, in the DiffServ model, predefined policies or SLAs are used to allocate the resources to a particular client. These policies are based on certain parameters such as the clients' peak traffic rate, the time for which the service is required, and the acceptable delay and jitter for an application. For many applications, for e.g. where the transmitted information is in the form of streaming media, the traffic rate is bursty in nature and is rarely at peak transfer rate, which is the amount of bandwidth allotted to the client. In such a situation, a portion of the allocated bandwidth remains unused. However, as this bandwidth is provisioned to the particular client, no other client can use it. Furthermore, bandwidth is reserved for a particular client and this reservation is applicable to all applications belonging to the client. As a result, a client will make reservations based on its maximum requirements, and every application belonging to the client, whether it is a streaming media application or just a simple mail application, will get this allocated service, which might result in delay or jitter being introduced into a client's dataflow for the streaming media application caused by the equal preferential treatment received by the other applications from the client.

In order to allocate these resources in a more intelligent fashion, a broker agent is used. The agent maintains a database of parameters pertaining to the various flows. Parameters such as service level agreements, current reservations/allocations, edge router configurations, service mappings/DSCP mappings, policy information, and management information. In accordance with these parameters the broker agent makes a

reservation for the client and assigns a DSCP for that service. This is certainly a better form of allocation, as sets of parameters are taken into consideration for each reservation. Each client gets to define his requirements and these get translated in to SLA's, which affects the resource allocation. But the end result is the same static reservation, where bandwidth once allotted to a client is used solely according to that client's traffic flow.

Thus arises a need to reuse the bandwidth wasted on each reservation that is made and if possible re-allot it to another client. The basic concept behind ARM is that by effectively knowing when a client is sending packets and how much of this allotted bandwidth is being used at any given time, the excess bandwidth can be reallocated without loss of promised service. Each client's request is equated to an SLS, which specifies the amount of bandwidth, the duration of the connection and a few other parameters. These parameters together map to a particular DSCP that is used to mark the incoming packets from that client so as to inform the routers to forward the packets with appropriate priority. In order to measure the traffic rate of every client, the bandwidth broker agent uses a meter that is provided by the DiffServ. For example, the TSW (Time Sliding Window) Tagger [10] is a meter that measures the average traffic rate, using a specified window size for the TSW2CM (Time Sliding Window 2 Color Marker) and TSW3CM (Time Sliding Window 3 Color Marker) [11] policers. With the knowledge of incoming traffic, different DSCP's are defined for various traffic rates. So when the broker agent notices a traffic rate that is less than the rate agreed upon, it steps down to a lower DSCP that suits the current rate. The remaining unused bandwidth is now sent to a pool of available bandwidth and is used when required by new clients or

then given back to clients when they require it. This helps us in increasing the number of clients who get the same kind of reservation guarantees, translating to more revenue for the same amount of bandwidth. It is a basic case of over allocation of resources. For the worst case scenario where all the clients send in traffic at their peak rate, the additional bandwidth is provided by dipping into the pool of bandwidth belonging to the best effort services. To achieve this, a set of preconditions for allocations is necessary. The preconditions are: 1. There should be a limit on the number of reservations allowed per class, and, 2. A fixed amount of bandwidth must be reserved for best effort services. By limiting the number of reservations for the Expedited Forwarding (EF), we can limit the number of premium service reservations, which have strict bandwidth requirements and correspondingly reduce the amount of unused bandwidth. Also by reserving bandwidth for best effort and providing a threshold of tolerance within which we can add or remove bandwidth as required, we efficiently reshuffle the unused bandwidth without adverse effect on the service agreements.

3.2 Illustrative examples

The functioning of the ARM algorithm is explained with the help of a test network. The test network includes two DiffServ domains, and a client belonging to a heterogeneous network architecture. The two DiffServ domains are required to show the inter-domain interaction between the broker agents to provide end-to-end resource allocation for a source-destination pair. The individual client from the heterogeneous network provides a method of verifying the allocation rules for out of domain traffic sources.

When a source1 that is outside the DiffServ domain requests service, it contacts the BB1 on DS1 enroute to the destination1. If the destination1 is within this domain, as shown in Figure1 using the black markings, the BB1 looks into its database, decides

upon the best available bandwidth, jitter and delay parameters, defines the SLA, and assigns a DSCP for the traffic flow between this source-destination pair. It then configures the edge router to mark the packets from this client with the correct DSCP so that the core routers just forward the packet according to the priority accorded to the flow.

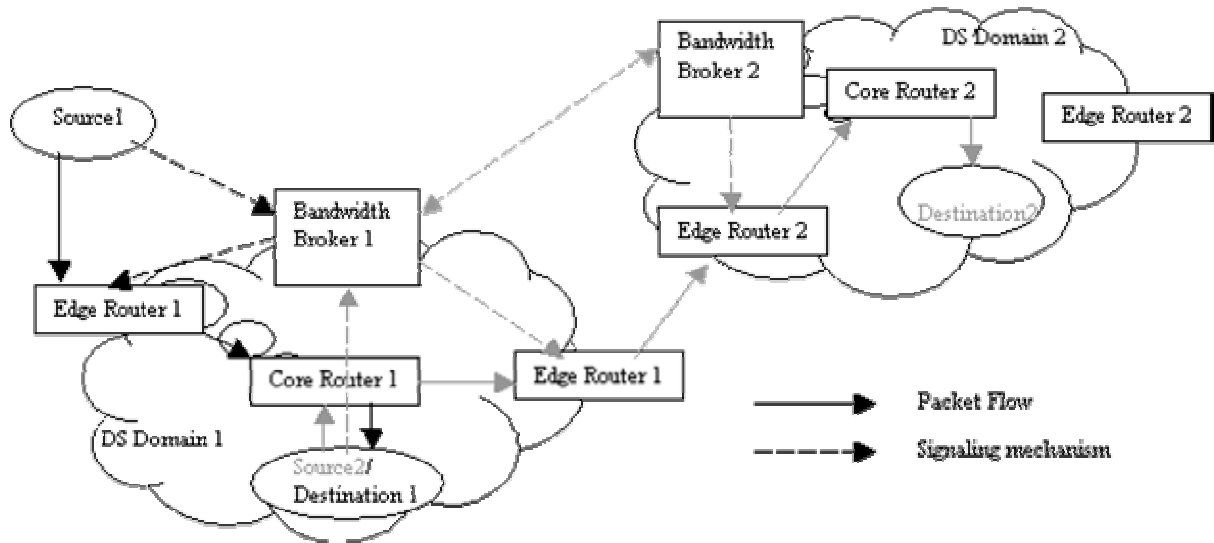


Figure 4: Test network showing different scenarios.

The BB1 also assigns a set of lower DSCPs, which define slightly lesser bandwidth requirements. During a particular packet exchange from source1, if the BB1 notices that it is running short of bandwidth to allocate, it uses the meter to check the traffic rate from the source, and if the rate is any lesser than the bandwidth allotted then it steps down the service to a lower DSCP which provides only the required amount of bandwidth, and the remaining unused bandwidth is returned to a pool of unused bandwidth which can be allotted to another client or returned to the client when needed. Supposing the source is in DS1 domain and the destination is DS2 domain, as shown by the gray markings, the source contacts the BB1 of its domain. The BB1 then looks at the

database and the routing table to figure out the downstream edge router and the peering BB2, and sends a RAR based on the SLA it has with the source. On confirming the reservation it allots a set of DSCPs corresponding to different levels of reservation, and configures the edge router of its domain to mark the packets from the source accordingly. The peering BB2 also configures its routers to perform the same kind of marking for that source destination pair. Both BBs define their own set of DSCPs for this flow. When there is need for conserving bandwidth in either domain, the corresponding BB decides to step down on the DSCP marking for the flow through that particular domain, thus saving bandwidth for reuse.

3.3 Implementation Using Network Simulator –2

We have implemented the ARM algorithm on the NS-2 toolkit [5]. The NS-2 (Network Simulator-2) toolkit has substantial functionality for simulating different network topologies and traffic models. NS also has an open architecture that allows users to add new functionalities which proves very useful for us. Along with the DiffServ patch provided by Nortel Networks, we can generate DS domains and create suitable test networks [6].

The DiffServ implementation has three modules to it. Two of them are with regards to the edge router and core routers, and the third module is the policy and resource manager. The policy class handles the creation, manipulation and enforcement of edge router policies. A policy defines the treatment the packets will receive at an edge router. Policies are set using Tcl [9] scripts. The policy class uses a policy table to store the parameter values. The table is in the form of an array of structures that has various fields such as SLA, current reservation, router configuration, policies, and DSCP mappings. The packet that arrives at the edge router is checked to decide as to which

traffic aggregate it belongs to, and a specified meter is used to check the average traffic rate of that client to make sure it corresponds to the current sending rate, else it gets downgraded to a lower DSCP.

The bandwidth broker is used to configure the policy module of the DiffServ. We define two modules for the broker agent. The modules are; user interface module through which the user/network operator can allocate resources, and a DiffServ manager, which does all the resource allocations. These allocations are reflected in the Policy module of the DiffServ, which is used to configure the edge routers.

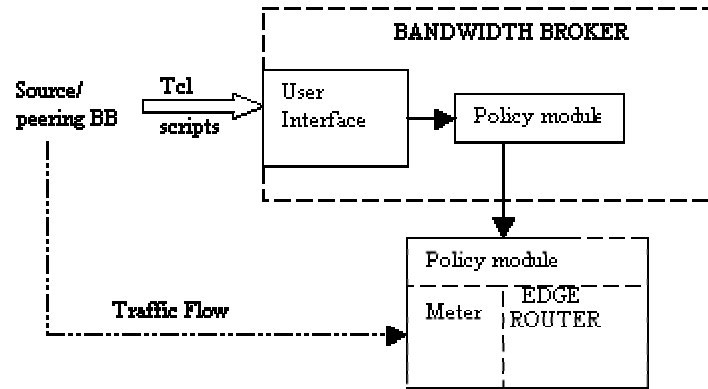


Figure 5: Modular breakup of the BB and its interactions in a traffic flow

The agent makes the provisioning based on the SLA's as agreed upon with the client/user (through the user interface module) using Tel scripts and in correspondence with other parameters in the database module such as the current reservations and the router configurations. The configuration changes are made to the policy module, and these changes are reflected in the policy module of the DS edge router. This achieves static provisioning. The Active Resource Management (ARM) algorithm keeps a track of every client's average traffic rate using a meter such as the TSW Tagger that is a part of the edge router. We use two meters, one that measures the traffic rate using a window

size and one that measures the current flow rate. The algorithm uses both the values to decide how much bandwidth can be reclaimed and re-allotted. Within the policy module we associate every source-destination flow with a policy type, meter type, current rate of traffic (the rate agreed upon with the client) and other policer specific parameters. We associate a set of DSCPs with this flow. Each DSCP corresponds to a different traffic rate, a lower traffic rate for each down step. When the algorithm measures a different traffic rate from the previous measurement, it moves the flow specifications to a different DSCP, which is configured with a traffic rate that is closer to the current traffic rate as indicated by the meter. The reclaimed bandwidth is used to accommodate more clients that require service guarantees.

Chapter 4

Experimentation and Evaluation

We have evaluated the ARM algorithm with three sets of experiments, each consisting of three comparisons. These sets of experiments are first performed on a DS domain that does the resource provisioning in its own capabilities (DS), then on a DS environment that uses a Bandwidth Broker to help provision the resources intelligently (DS+BB), and then finally on the DS environment that uses bandwidth brokers implementing the ARM algorithm (DS+BB+ARM). The first experiment allocates the entire available bandwidth, while the second experiment pushes the allocation over the limit, and finally the third experiment tests the system for an increased duration of simulation time.

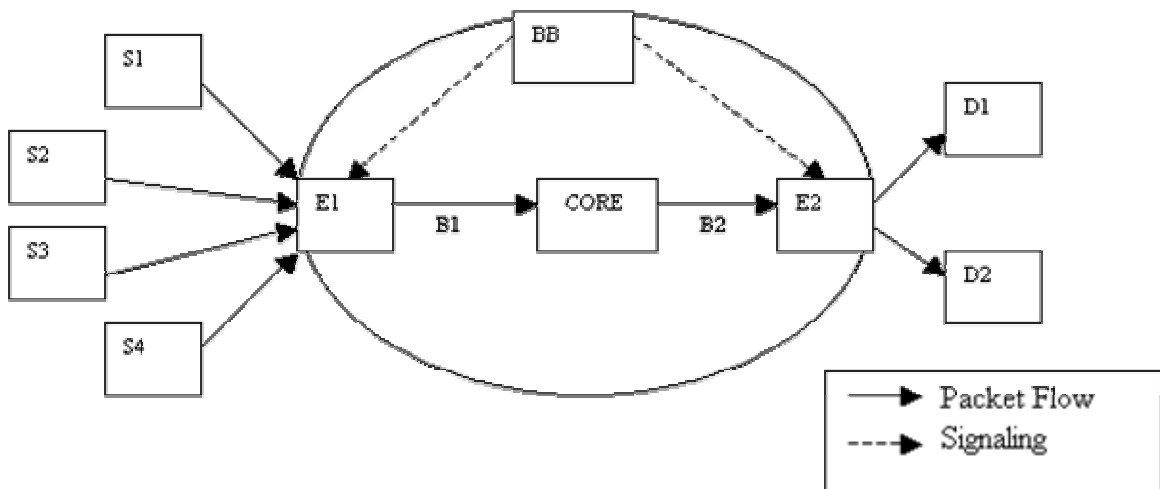


Figure 6: Test network

Our test network is shown in Figure 6 and consists of 5 nodes, S1, S2, S3, D12 and D3, 3 DS enabled routers, E1 and E2 are the edge routers, and Core is a core router. The DS

Domain consists of the three routers and one bandwidth broker agent, which configure the edge routers. The table below explains the various entities with their parameters.

Name	Type	Parameter
S1	Source 1	Bursty Source with Pareto distribution. Pkt size 500, burst time 500ms, idle time 300ms, rate 1Mbps.
S2	Source 2	Bursty Source with Exponential Pkt size 500, burst time 500ms, idle time 300ms, rate 1Mbps
S3	Source 3	CBR type Source on UDP. Rate 1.5Mbps.
S4	Source 4	FTP application on TCP. It produces packets at regular intervals.
D1	Destination 1	Generic node
D2	Destination 2	Generic node
BB	Bandwidth Broker agent	Configures the Edge routers.
B1 & R2	Link Bandwidth	Link Bandwidth of 5Mbps between E1 & Core & E2

Table 2: Legend for test network.

The source parameters are Committed Information Rate (CIR), Alternate Committed Information Rate (ALTCIR), Peak Information Rate (PIR), Alternate Peak Information Rate (ALTPIR), Policer for first set of parameters (POL) and Policer for alternate set of parameters (ALTPOL).

Source	CIR	PIR	POL	ALTCIR	ALTPIR	ALTPOL
S1	1Mbps	2Mbps	EF	750Kbps	1.5Mbps	EF
S2	1Mbps	2Mbps	EF	750Kbps	1.5Mbps	EF
S3	1.5Mbps	3Mbps	TSW2CM	1Mbps	2Mbps	TSW3CM
S4	1.5Mbps	3Mbps	TSW3CM	1Mbps	2Mbps	TSW3CM

Table 3: Initial policy request

4.1 Experiments:

4.1.1 Experiment 1: Exact Allocation of Resources.

In this experiment we test the DiffServ's capability to provide service when all the bandwidth is used up. Both the BB experiment and the ARM algorithm experiment allocate less than maximum bandwidth available and result in a better utilization of bandwidth. Shown below (Table 4) are policy tables of the three evaluations DS, DS+BB, and DS+BB+ARM. The bandwidth used in each case is calculated by adding the CIR. Evaluation 3, using the ARM algorithm, shows two sets of policy tables. The first table corresponds to the initial allocation made, and the second table shows the resultant run time allocations made by the ARM algorithm.

1.A: DS simulation results:

Flow	Policer Type	Initial Codepoint	CIR	PIR
S1 to D1	EF	10	500Kbps	500Kbps
S2 to D1	EF	11	1Mbps	1Mbps
S3 to D2	TSW2CM	15	1.5Mbps	----
S4 to D2	TSW3CM	17	2Mbps	4Mbps

1.B: Broker simulation results:

Flow	Policer Type	Initial Codepoint	CIR	PIR
S1 to D1	EF	10	500Kbps	500Kbps
S2 to D1	EF	11	1Mbps	1Mbps
S3 to D2	TSW2CM	15	1.5Mbps	----
S4 to D2	TSW3CM	17	1.5Mbps	3Mbps

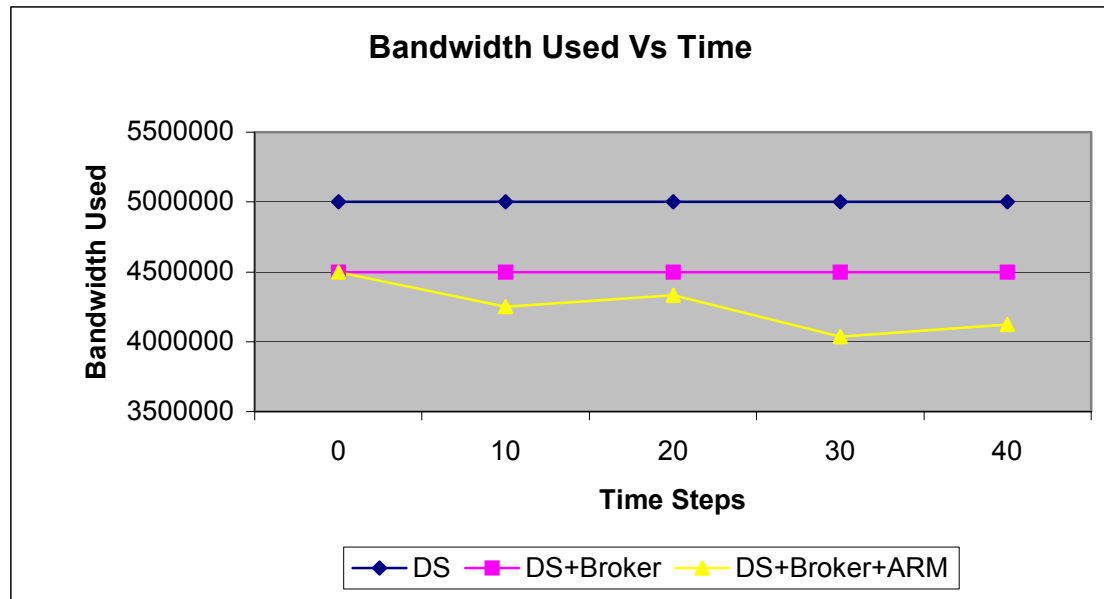
Table 4: Policy Table for Experiment 1.

1.C: ARM algorithm simulation results:

Flow	Policer Type	Initial Codepoint	CIR	PIR
S1 to D1	EF	10	500Kbps	500Kbps
S2 to D1	EF	11	1Mbps	1Mbps
S3 to D2	TSW2CM	15	1.5Mbps	----
S4 to D2	TSW3CM	17	1.5Mbps	3Mbps
Flow	Policer Type	Initial Codepoint	CIR	PIR
S1 to D1	EF	10	375.0 Kbps	500Kbps
S2 to D1	EF	11	750.0Kbps	1Mbps
S3 to D2	TSW2CM	15	1.5Mbps	----
S4 to D2	TSW3CM	17	1.496Mbps	2Mbps

Table 4: Policy Table (continued).

The ARM algorithm shows improved bandwidth utilization and we realize a conservation of more than 50% of the bandwidth. A graphical representation of the above tabular values is shown below.



Graph 1: A plot of bandwidth used Vs time for Experiment 1.

The next table of packet statistics shows evidence that not only have we managed to optimize bandwidth utilization, but also have a better mechanism to manage traffic flow and reduce the number of dropped packets.

1.A. DS Packet Statistics

CP	TotPkts	TxPkts	Ldrops	edrops
All	83051	79482	3518	51
10	4472	4472	0	0
11	12303	12303	0	0
15	25056	23799	1257	0
16	8026	7566	409	51
17	33083	31231	1852	0
18	111	111	0	0

1.B. Broker Packet Statistics.

CP	TotPkts	TxPkts	Ldrops	edrops
All	71989	71989	0	0
10	5481	5481	0	0
11	10838	10838	0	0
15	15142	15142	0	0
16	4767	4767	0	0
17	35659	35659	0	0
18	102	102	0	0

1.C. ARM Packet Statistics

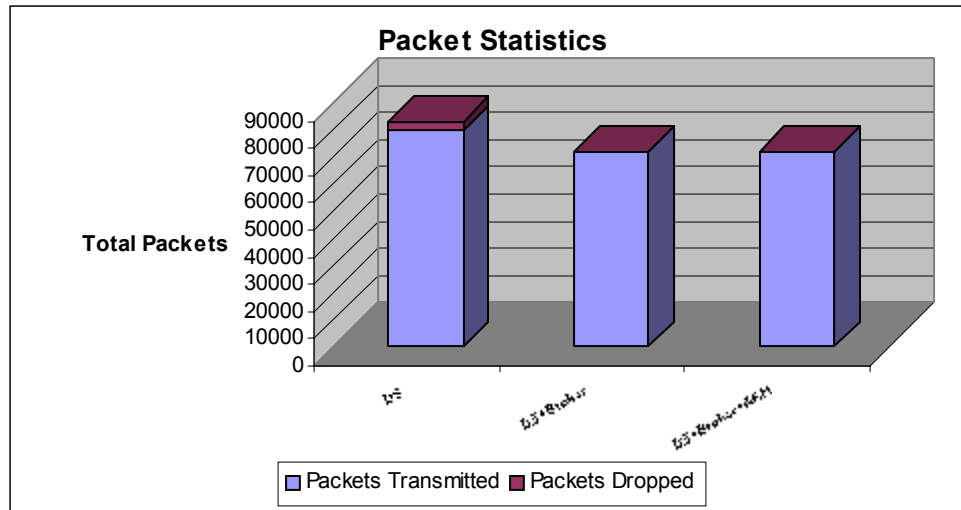
CP	TotPkts	TxPkts	Ldrops	edrops
All	71989	71989	0	0
10	5481	5481	0	0
11	10838	10838	0	0
15	15142	15142	0	0
16	4767	4767	0	0
17	35631	35631	0	0
18	130	130	0	0

Key:

CP	Codepoints
TotPkts	Total Packets
TxPkts	Transmitted Packets
Ldrops	Late drops
Edrops	Early drops

Table 5: Packet Statistics for Experiment 1

A graphical representation of the packet statistics is shown below. It is clearly visible that while using the ARM algorithm, the service guarantees are maintained, and no packets are dropped.



Graph 2: Packet statistics for Experiment 1.

4.1.2 Experiment 2: Over Allocation of Resources

In the second experiment, we stress the allocation limits by over allocating. This is achieved by increasing Source 2 requirements to 2Mbps. The Broker manages to keep the allocation under control and subsequently the ARM algorithm improves upon the broker's allocation. Experiments 2.A, 2.B and 2.C show the DS results, broker results and the ARM algorithm results respectively.

2.A: DS simulation results:

Flow	Policer Type	Initial Codepoint	CIR	PIR
S1 to D1	EF	10	500Kbps	500Kbps
S2 to D1	EF	11	1.5Mbps	1.5Mbps
S3 to D2	TSW2CM	15	1.5Mbps	----
S4 to D2	TSW3CM	17	2Mbps	3Mbps

2.B: Broker simulation results:

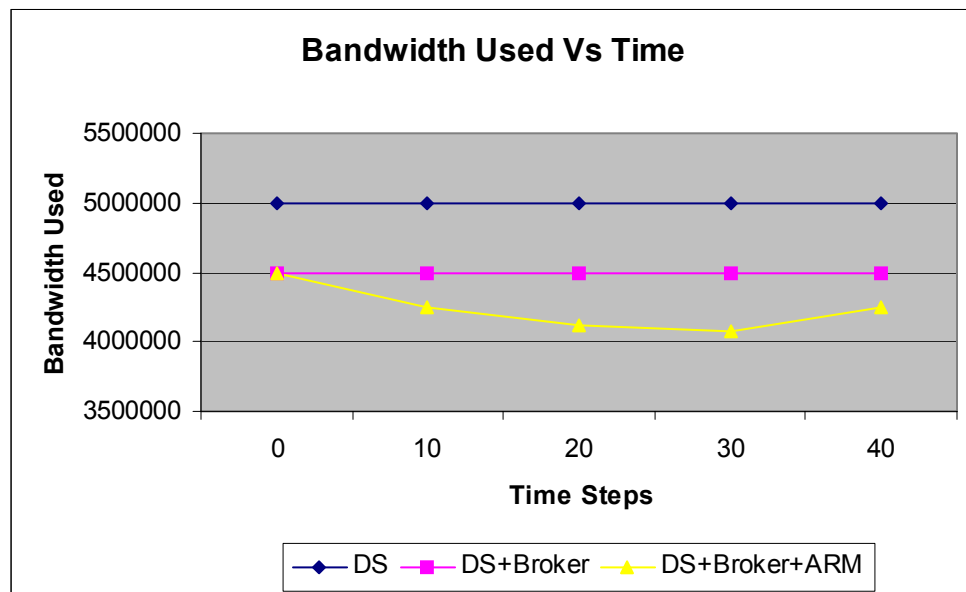
Flow	Policer Type	Initial Codepoint	CIR	PIR
S1 to D1	EF	10	500Kbps	500Kbps
S2 to D1	EF	11	1Mbps	1Mbps
S3 to D2	TSW2CM	15	1.5Mbps	----
S4 to D2	TSW3CM	17	1.5Mbps	3Mbps

2.C: ARM algorithm simulation results:

Flow	Policer Type	Initial Codepoint	CIR	PIR
S1 to D1	EF	10	500Kbps	500Kbps
S2 to D1	EF	11	1Mbps	1Mbps
S3 to D2	TSW2CM	15	1.5Mbps	----
S4 to D2	TSW3CM	17	1.5Mbps	3Mbps
Flow	Policer Type	Initial Codepoint	CIR	PIR
S1 to D1	EF	10	500Kbps	500Kbps
S2 to D1	EF	11	750.0Kbps	1Mbps
S3 to D2	TSW2CM	15	1.497Mbps	----
S4 to D2	TSW3CM	17	1.498Mbps	3Mbps

Table 6: Policy Table for Experiment 2.

The graphical representation of the bandwidth used over the 40 timesteps duration that the experiment was conducted for is shown below. Some used bandwidth was reclaimed for further use, thus increasing the number of clients requiring guaranteed service.



Graph 3: A plot of bandwidth used Vs time for Experiment 2.

The packet statistics for over allocation are as shown below. Even during over allocation the ARM algorithm manages to maintain the service levels that have been guaranteed to the customers.

1.A. DS Packet Statistics

CP	TotPkts	TxPkts	Ldrops	edrops
All	62292	53843	8448	1
10	9195	9195	0	0
11	14564	14564	0	0
15	19235	14998	4237	0
16	38	37	1	0
17	19190	14983	4207	0
18	70	66	3	1

1.B. Broker Packet Statistics.

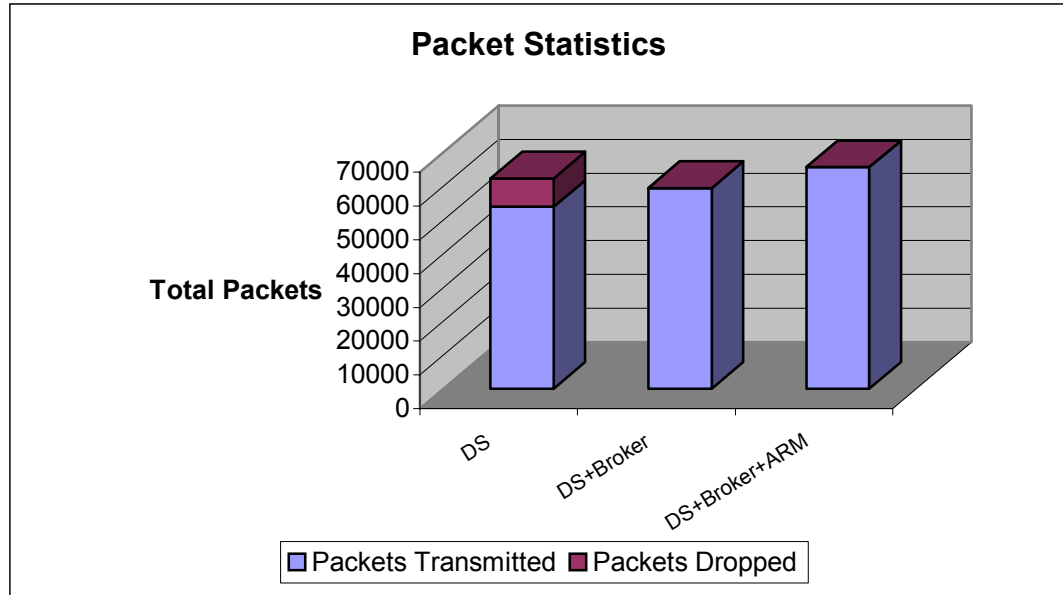
CP	TotPkts	TxPkts	Ldrops	edrops
All	59410	59410	0	0
10	9827	9827	0	0
11	5811	5811	0	0
15	19935	19935	0	0
16	86	86	0	0
17	23713	23713	0	0
18	38	38	0	0

1.C. ARM Packet Statistics

CP	TotPkts	TxPkts	Ldrops	Edrops
All	65550	65550	0	0
10	3972	3972	0	0
11	889	889	0	0
15	24904	24904	0	0
16	74	74	0	0
17	35610	35610	0	0
18	101	101	0	0

Table 7: Packet Statistics for Experiment 2.

The graphical representations of the packet statistics are shown below.

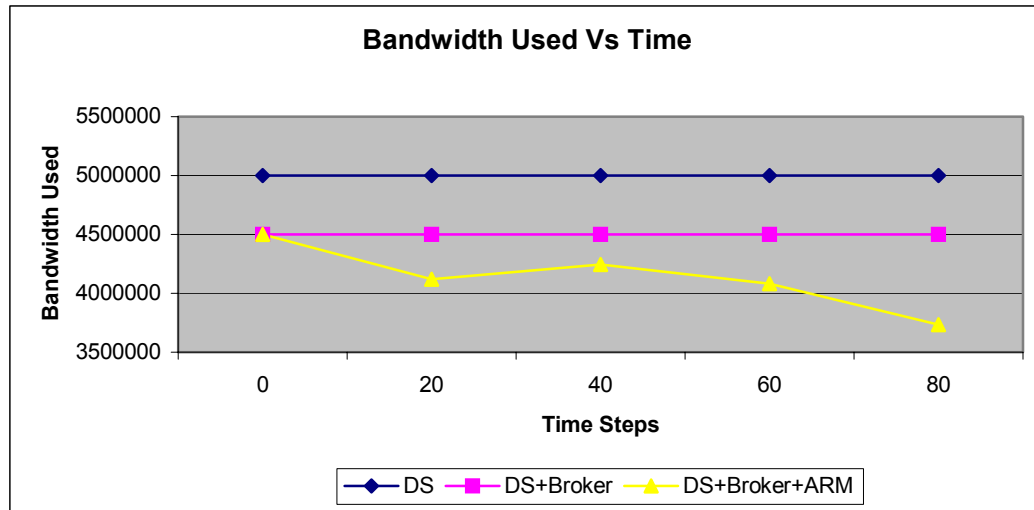


Graph 4: Packet statistics for Experiment 3.

4.1.3 Experiment 3: Over Allocation for an Extended Period of Time

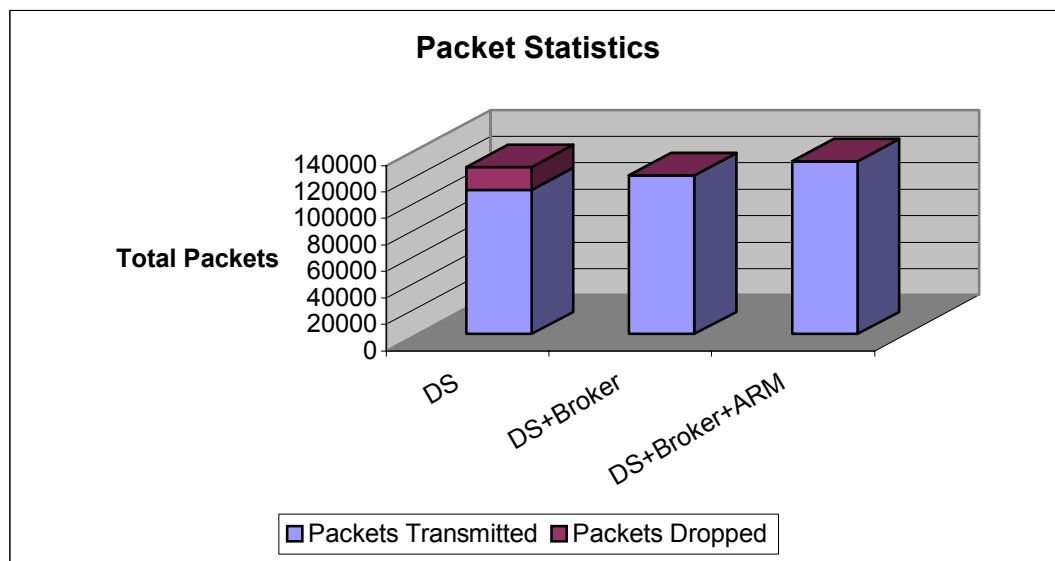
For the third experiment, we extended the second experiment to stress the system by doubling the period of simulation to 80.0 timesteps. It can be seen that, as the traffic increases, the broker results and the ARM algorithm are still within limits of the guarantees given, while the allocation maintains the same improved performance. This is because of the improved queue scheduling mechanism we use called Priority – WRR scheduling, which lets us define one queue as priority queue and the rest of the queues as weighted round robin queues.

In the graph below we can see the bandwidth usage against the increased duration of the experiment.



Graph 5: Plot of Bandwidth Used Vs Time for Experiment 3.

In the last graph, the packet statistics for the experiment shows how the improved queuing mechanism helps prevent packet drops.



Graph 6: Packet Statistics for Experiment 3.

Chapter 5

Summary and Future Work

5.1 Summary

There is a need for guaranteed services for real time media and mission critical traffic that cannot be provided by standard IP methods. The Differentiated Services framework provides a suitable, scalable and less complex means for providing these guarantees and with the help of the bandwidth broker agent, a level of intelligent resource provisioning is achieved. But these methods cause wastage of bandwidth due to the static reservation scheme followed by them. We strive to reach a level of optimization of these resources by introducing the Active Resource Management algorithm, an algorithm that reallocates the unused bandwidth reserved for specific clients when not used by them, to other clients, thus providing optimum usage of the limited bandwidth that is available. We have implemented ARM using NS-2 and evaluated it with promising results. We were able to save up on at least 25% of the bandwidth allocated to the individual flows.

5.2 Future Work

Further work in this area of research is:

1. Providing resource reservation on a per application basis, rather than the current per client basis by using a form of content aware bandwidth broker. Since all applications used by a client receive the guarantees requested by the client, packets from all of them go to the same priority queue. In times of congestion, there is a possibility that the packets of the multimedia application requiring the QoS could be dropped, while a simple mail application from the same client could have its packets be delivered without loss of service.

2. The broker architecture with the ARM algorithm can be made into a middleware application that can provide resource management to any heterogeneous environment, and not be restricted to the DiffServ.

References

1. Andrew S. Tanenbaum, *Computer Networks*, Third Edition, Prentice Hall, 1996.
2. Chris Metz, "IP QoS: traveling in the first class on the Internet". *Proceedings of IEEE Internet Computing*, Vol. 3, No. 2, March/April 1999.
3. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. "An Architecture for Differentiated Services". RFC 2475. December 1998.
4. K. Fall and K. Vardhan, editors. "NS notes and documentation". Computer Science Department, University of California – Berkeley.
<http://www.isi.edu/nsnam/ns/ns-documentation.html>
5. The network simulator (NS-2) Homepage. <http://www.isi.edu/nsnam/ns/>
6. P. Piedad, J. Ethridge, M. Baines and F. Shallwani. "A Network Simulator Differentiated Services Implementation". Open IP, Nortel Networks.
<http://www7.nortel.com:8080/CTL/>
7. K. Nichols, V. Jacobson and L. Zhang. "A Two Bit Differentiated Services Architecture For The Internet". RFC 2638, July 1999.
8. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. "Assured Forwarding PHB Group". RFC 2597. June 1999.
9. Brent B. Welsh, *Practical programming in Tcl and Tk*. Third Edition, Prentice Hall, 1999. (ISBN 0130220280)
10. D. Clark, and W. Fang. "Explicit Allocation of Best Effort Packet Delivery Service". *IEEE/ACM Transactions on Networking*, 6(4): 362-373, August 1998.
11. W. Fang, N. Seddigh, and B. Nandy. "A Time Sliding Window Three Color Marker". Internet draft, March 2000.
12. J. Heinanen, T. Finland, and R. Guerin. "A Single Rate Three Color Marker". Internet draft, May 1999.
13. J. Heinanen, T. Finland, and R. Guerin. "A Two Rate Three Color Marker". Internet draft, May 1999.
14. Qbone signaling design team. <http://qbone.internet2.edu/bb/>
15. Jun Ogawa and Yuji Nomura. "A Simple Resource Management Architecture for Differentiated Services". *Proceedings of Internet Society (INET '00)*, 315-438, June 2000.
http://www.isoc.org/isoc/conferences/inet/00/cdproceedings/1h/1h_1.htm

16. Francis Reichmeyer, Lyndon Ong, Andreas Terzis, Lixia Zhang and Raj Yavatkar. "A Two-Tier Resource Management Model for Differentiated Services Networks". Internet Draft, November 1998.
17. Róbert Szabó, Tamás Henk, Vlora Rexhepi, Georgios Karagiannis. "Resource Management in Differentiated Services (RMD) IP Networks". *Proceedings of the International Conference on Emerging Telecommunications Technologies and Applications, ICETA 2001*, 64- 80, Kosice, Slovak Republic, October 2001.
18. R. Braden, D. Clark, S. Shenker. "Integrated Services in the Internet Architecture: an Overview", IETF RFC-1633, June 1994
19. S. Blake, D. Black, M. Carlson, E. Davies, Z. Whang, W. Weiss, "An Architecture for Differentiated Services", IETF RFC-2475, December 1998.
20. B. Braden, B. Ed., et al., "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification", IETF RFC-2205, September 1997
21. G. Fehér, K. Németh et al., "Boomerang - A Simple Protocol for Resource Reservation in IP Networks". *Proceedings of IEEE Workshop on QoS Support for Real-Time Internet Applications*, 240-264, Vancouver - Canada, June 1999.
22. Ping Pan and Henning Schulzrinne, "YESSIR: a simple reservation mechanism for the Internet", *ACM Computer Communication Review*, vol. 29, no.2, pp. 89--101, April 1999.
23. W. Almesberger, T. Ferrari, J. -Y. Le Boudec, "SRP: a Scalable Resource Reservation Protocol for the Internet". *ACM Computer Communications Review*, vol. 21, no. 14, pp. 1200-1211, September 1998.
24. A. Eriksson, "Resource reservation in a connectionless network". *Proceedings of Performance Information and Communication Systems (PICS'98)*, May 1998.
25. Intel®, Policy-Based Network Management (PBNM),
<http://www.intel.com/labs/manage/pbnm/index.htm>
26. Ian Foster, Carl Kesselman, Craig Lee, Bob Lindell, Klara Nahhrstedt, Alain Roy, "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocations". *Proceedings of the Globus Retreat (IWQoS '99)*, June 1999.
<http://www.globus.org/documentation/incoming/iwqos.pdf>
27. Alain Roy, "End to End Quality of Service for High-End Applications". PhD Dissertation, August 2001. <http://www.cs.wisc.edu/~roy/publications/>
28. G. de Veciana, G. Kesidis and J. Walrand, "Resource management in wide-area ATM networks using effective bandwidths". *IEEE Journal on Selected Areas in Communications*, 1081-1090, May 1995.
29. Croll, Alistair and Packman, Eric. *Managing Bandwidth: Deploying Across Enterprise Networks*. Prentice Hall PTR, January 2000. (ISBN 0130113913)

30. Kalevi Kilkki, *Differentiated Services for the Internet*, New Riders Publishing, June 1999. (ISBN 1578701325)
31. Chuck Semeria, "Supporting Differentiated Service Classes: Active Queue Memory Management", White Paper, Juniper Networks Inc.
<http://www.juniper.net/techcenter/techpapers/200021.html>
32. Satishkrishnan Gopalakrishnan, "Trio - A scheme for active resource management for the DiffServ", PhD Dissertation, University of Texas, A&M, May 2000.
33. Andreas Terzis, Lan Wang, Lixia Zhang, "A Scalable Resource Management Framework for Differentiated Services Internet". *Proceedings of 3rd NASA/NREN Workshop on QoS for the Next Generation Internet*, 128-145, August 1998.
34. Anna Charny, "EF PHB Redefined". Internet Draft, November 2000.
35. Differentiated Services testing at TF-TANT. <http://www.cnaf.infn.it/~ferrari/tfng/ds/>
36. CSIRO and Australian Academic and Research Network (AARNET) testbed.
http://www-networks.tip.csiro.au/~mminhazu/dif_serv.html
37. Massachusetts Institute of Technology (MIT) Differentiated Services page.
<http://diffserv.lcs.mit.edu/>
38. University of Kansas IPQoS project. <http://qos.ittc.ukans.edu/>
39. G. Hoo, W. Johnston, I. Foster, and A. Roy. "QoS as middleware: Bandwidth broker system design". *Poster session for the 8th High Performance Distributed Computing conference (HPDC '99)*, Redondo Beach, California, August 1999.
40. Policy Based Network Management Software by Orchestream.
<http://www.orchestream.com>
41. Policy Based Bandwidth control and application prioritization software by Extremeware. <http://www.extremenetworks.com/products/datasheets/entmng.asp>
42. Simple Network Management Protocol Homepage. <http://www.snmp.com>
43. D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry. "The COPS (Common Open Policy Service) protocol", Internet Draft 2748, January 2000.