The Survey of Bandwidth Broker

Shaleeza Sohail and Sanjay Jha
School of Computer Science and Engineering
The University of New South Wales, Sydney 2052, Australia
Email: {sohails,sjha}@cse.unsw.edu.au

UNSW-CSE-TR-0206

May 2002

THE UNIVERSITY OF
NEW SOUTH WALES

SYDNEY • AUSTRALIA

Abstract

Keeping in mind the present network management research trends, it can be safely stated that in the near future enterprise networks and ISPs will need a network management entity to dynamically manage QoS networks. DiffServ is one of the emerging networks that introduces bandwidth broker as its logical resource, network and policy management module. Due to the complex and huge functionality provided by bandwidth broker, it has very large number of semi explored research areas. This survey is an effort to briefly discuss some of the developments in the ongoing process of defining and implementing a functional bandwidth broker.

1 Introduction

In order to support Quality of Service (QoS) in the network, new architecture such as Intserv and Diffserv have been proposed in the IETF. These architectures support diverse service levels for multimedia and real-time applications. DiffServ architecture is capable of providing well defined end-to-end service over concatenated chains of separately administered domain by enforcing the aggregate traffic contracts between domains. At the interdomain boundaries, service level agreements (SLAs) specify the transit service to be given to each aggregate. SLAs are complex business related contracts that cover a wide range of issues, including network availability guarantees, payment models and other legal and business necessities. SLA contains a Service Level Specification (SLS) that characterizes aggregates traffic profile and the PHB to be applied to each aggregate. To automate the process of SLS negotiation, admission control and configuration of network devices correctly to support the provisioned QoS, each DiffServ network may be added with a new component called a Bandwidth Broker (BB) [32].

An ISP, for example, can dynamically negotiate different service level agreements and bandwidth guarantee with a given customer. Alternatively, a server provider could charge different rates for bandwidth depending on the demand. To do this, the bandwidth broker will contain the ability, using standards based protocols, to communicate with remote bandwidth broker in order to negotiate the Service Level Specification and with local enforcers to determine the state of the network as well as configure the network. The bandwidth broker will take into consideration the ability of the entire network to deliver the policy request. Suppose that a customer requests that its minimum bandwidth guarantees be increased from 2Mbps to 5Mbps. The bandwidth broker will check the state of the network over a period as well as the number of other commitments that have been made before making any decision.

As we can see, the bandwidth broker is a complex entity that might need integration of several technologies such as standard interface for inter/intra domain communication, protocol entity for communication, standard protocol and database. Organisational policies can be configured by using the mechanism provided by BB. On the inter domain level BB is responsible of negotiating QoS parameters and setting up bilateral agreements with neighbouring domains. On intra domain level BB's responsibilities include configuration of edge routers to enforce resource allocation and admission control. With the help of Simulation [25], it has also been suggested that bandwidth broker in DiffServ architecture can be effectively used to provide QoS to real time applications like VoIP. Moreover these studies also indicate that admission control mechanism of BB improves the profit for the ISPs by improving network resource utilisation.

Different aspects of BB are discussed in detail in the following sections.

In section 2, the role of BB in DiffServ domain is described. In section 3, the architecture of BB is elaborated. Section 4 has some of the related ongoing research work in this field. Section 5 introduces distributed BB architecture. Section 6 discusses pricing related issues. Section 7 investigates advance research topics in the field of SLAs related to BB. Active resource management architecture is discussed in section 8. BBs extended functionality in networks other than DiffServ and in hybrid networks is discussed briefly in section 9. A discussion about future of research in the field of BB in section 10 concludes the survey.

2 Bandwidth Broker in DiffServ

The functional model of BB in DiffServ domain is discussed in this section. DiffServ architecture provides a simple mechanism to provide QoS to the network traffic. The main functionality of the DiffServ architecture is in the edge routers, core routers are maintained as simple as possible. The traffic entering a DiffServ domain is classified and conditioned at the boundary of the network and then assign to different behaviour aggregates (BA). The flows entering a domain are classified into one of many classes based upon the value of DiffServ code point(DSCP) in the header of the packet[18]. All packets having same DSCP are treated in the same manner and they belong to same behaviour aggregate(BA). The core routers simply forward the packet according to the treatment it deserves on the basis of its BA.

The main resource management entity in DiffServ domain is BB. BB maintains policies and negotiates SLAs with customers and neighbouring domains. The interaction of BB with other components of DiffServ domain as well as the end-to-end communication process in DiffServ domain is shown in the Figure1. The figure shows that when a flow needs to enter the DiffServ domain or a local user wants to send some traffic, BB is requested to check related SLA. BB is responsible for admission control as it has global knowledge of network topology and resource allocation. BB decides as to allow the traffic or not on the basis of previously negotiated SLAs. In case of a new flow BB might have to negotiate a new SLA with the neighbouring domain depending upon the traffic requirements. Once BB allows the traffic, the edge router or leaf router needs to be reconfigured by BB. SLA negotiation is a dynamic process due to the ever changing requirements of the network traffic.

3 Bandwidth Broker Architecture

The previous section discussed the functionality provided by BB in a DiffServ domain. The last section mentions that BB treats resource allocation requests and in the process negotiates SLAs with other domains as well as

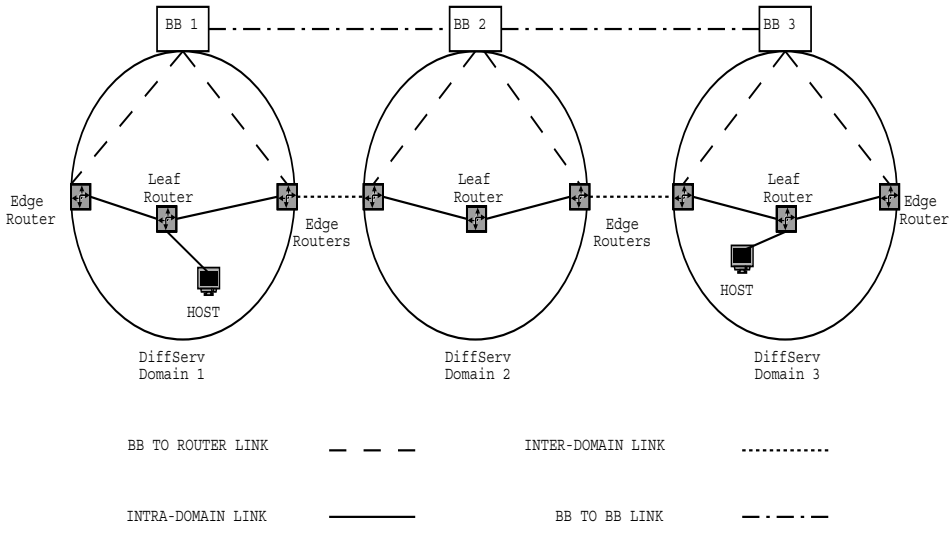

Figure 1: Role of BB in DiffServ

customers. Moreover BB maintains a policy and resource database to have up-to-date knowledge of the domain, essential to make resource allocation decisions, BB also configures edge routers to enforce the policy decisions. As the BB is a complex entity, it needs several components. This section describes the basic components of a BB as shown in Figure 2.

These components are categorised in[31] as follows:

1. Data interface
    - Routing tables
    - Data repository
2. Key protocols
    - User/application protocols
    - Inter domain communication protocols
    - Intra domain communication protocols

In the following sections each of these components are separately discussed.

3.1 Data Interface

BB as being the main resource management entity in DiffServ domain keeps a database of all management related issues. The routing table database is

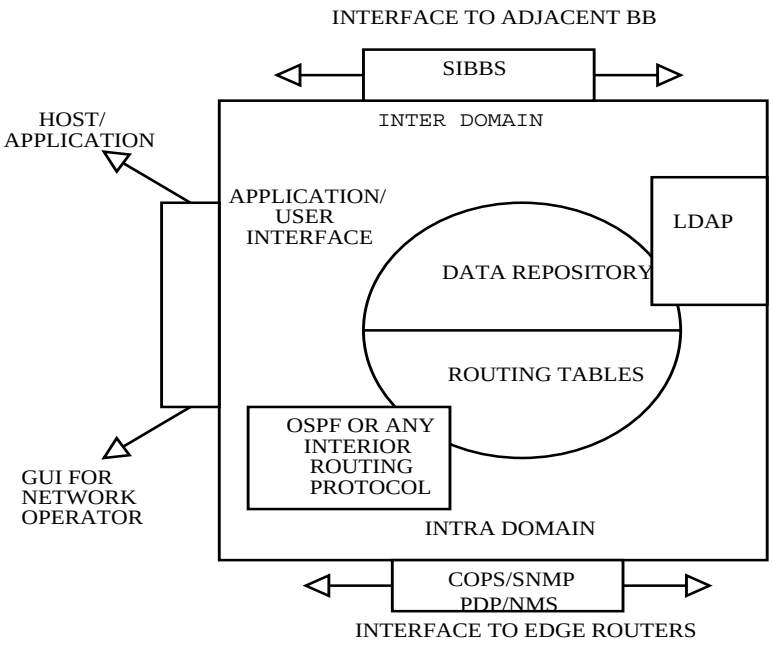

Figure 2: Architecture of Bandwidth Broker

maintained to provide complete network picture for the process of resource allocation. The SLA, policy and resource database maintained, enforces policy rules in all the decisions taken by BB. Both of these databases are discussed separately in the following subsections.

3.1.1 Routing Tables

BB maintains a complete topological map of its domain in order to determine

- The egress/ingress router for any flow in its domain;
- Next domains in the path of a flow towards the destination in case the destination is in another domain;
- First leaf router in case the request is generated by any host in its own domain.

The best way to capture an up-to-date routing table is by configuring the intra domain routing protocol on BB. BB appears as one of the nodes of the network to all other network devices as well as to the routing protocol. The very nature of the routing protocol enables BB to capture topological as well as routing information of the complete network. Different protocols like OSPF, RIP v1, RIP v2 etc can be easily implemented in this scenario depending upon the network administrator's preference.

3.1.2 Data Repository

To provide QoS in the network, BB must have a comprehensive picture of complete network. In general the areas about which BB maintains information are policy, SLA, network management and current resource allocation status [31]. Routers can also be configured to provide monitoring data to enhance the security of the network and optimal resource usage. Router's configuration data and information about BB's own components is maintained for the purpose of fault tolerance.

An interface is required to access the database by BB. The most common and easy to implement protocol in this scenario is light weight directory access protocol(LDAP).

LDAP is an open standard protocol for accessing information services. The use of LDAP in this scenario can be justified by its light weight nature and simplicity of implementation. It uses TCP/IP stack and can be easily used to access stand alone directory services. LDAP encodes many protocol elements as ordinary strings and light weight BER encoding is used to encode all protocol elements. LDAP is based on client/server model, in which server is responsible for handling all referrals and there is no need to send these referrals to clients. Moreover server is responsible for directory access when it receives a request from client.

LDAP is specially designed to access information stored in directories. The main reason to use LDAP in this case is due to the fact that it is designed to store information that is more read then write. Relational database management system (RDBMS) can also be used. However, the LDAP has the following advantages over the use of RDBMS in storing policies:

- IETF working groups has given a standard LDAP schema for policies [5];
- Conforming to the LDAP schema makes the policy database standards compliant which can then be read by any third party LDAP client;
- The model of LDAP, optimising few writes with a lot of reads works well for policy storage which will not be modified very often but can be assumed to be read very often.

LDAP has few limitations too, specially in case of complex policies and security. LDAP does not support very complex policies as it assumes that policies are stored as directories. Moreover in case of multiple clients, the transaction mechanism is not optimal. LDAP is improving with time and the latest LDAP version 3 eliminates few security drawbacks.

For policy based management Policy Information Base(PIB) is used as an information repository to keep policy related information. More description of PIB is provided with COPS-PR[13] [5] in the next section.

3.2 Key Protocols

BB performs complex functions in a DiffServ domain and it needs a set of protocols to interact with different components of the domain. In the following subsections some of the prominent protocols used by BB are discussed briefly.

3.2.1 User/Application Protocols

There is a need for a protocol or interface for network operator and user/application to interact with the bandwidth broker. The network operator may use this interface to monitor or update the performance related features of BB. The user/application requires the interface or protocol to request or query the BB. In return, the resource allocation requests (RAR) from user needs a response from BB to assure the QoS promised in SLAs.

3.2.2 Intradomain Communication Protocols

The intra domain protocol used in the DiffServ domain is of local significance to the network administrator. However in general, all intra domain protocols used in DiffServ domain should be capable of transferring configuration parameters from BB to edge routers. Every router has a builtin mechanism for its configuration from the manufacturer. COPS, SNMP and Telnet are three mechanisms which exists today to configure routers, but not all routers support all of these mechanisms.

Common open policy services (COPS) is the protocol standardised by IETF Resource allocation protocol (RAP) working group [15]. COPS is used to send policy decisions from policy decision point(PDP) to policy enforcement point(PEP). PEP has the ability to handle IP traffic and implements policy based admission controls for data flows, whereas PDP has complete view of network and configures its PEPs according to the network policies. BB is suppose to have the functionality of PDP and all the edge routers are configured as PEPs. COPS is a client/server protocol, where server(PDP) has a TCP connection with all its clients(PEP), so there is no need of reliability mechanism in the protocol itself.

COPS supports two models: outsourcing and policy provisioning. In outsourcing model, a PEP can outsource its decision by querying PDP and waiting for PDP's decision before communicating its own decision. Outsourcing model is best suited for IntServ/RSVP scenario. The policy provisioning model does not support any correlation among PEP's query and PDP's response. The DiffServ architecture uses policy provisioning model.

COPS has some features that makes it suitable for using with BB. COPS has keep alive mechanism by which PEP knows that its PDP is up or down. COPS has the option for PDP to redirect a client if it does not support that client type or for load balancing. As QoS enabled services are highly

vulnerable to denial and theft of services, COPS uses IPSec and integrity objects to provide the required security.

For support of policy provisioning a new client type COPS for provisioning (COPS-PR) is introduced in [15]. It is independent of the type of policy being provisioned as it can be QoS or security. COPS-PR has support for real time event driven communication mechanism. PEP has only one connection to PDP in one area of policy control, it supports large atomic transactions of data and efficient error reporting. It has state sharing/synchronisation and exchange differential updates only. On the time of bootup the PEP establishes a connection with PDP and sends all device relevant information. PDP replies with all provisioned policies that are relevant to the device. In case there is some change in policies at PDP then it sends update message and if there is some change at PEPs end then it sends the changes to PDP which can reply with new relevant policy provisioning elements.

The RAP working group is is presently working on available COPS objects specially COPS-PR. Defining data definition language for COPS-PR and standardising architecture of COPS based management are few of its main work items[3].

Policy Information Base(PIB) is actually a database for policy information. PIB is defined to be used with COPS-PR[13] [5] discussed in next section. PIB stores policy provisioning instances in a tree structure where the branches can be visualised as representing policy rules or Policy Rule Classes(PRCs) and the leaves can be seen as representing contents of policy rules or Policy Rules Instances(PRIs). PRIs are identified by Provisioning Instance Identifier(PRIDs). Every role of each policy enforcement point (PEP) is stored in the PIB as an instance and the roles are unique in their nature so PRID is a unique name in a COPS object. An example of PIB tree numbering of PRID is "5.4.3.2.1". PRC is represented by first four digits "5.4.3.2" and the PRI is identified by last digit "1". Schema is the high level static definition of network policy in the shape of directories. PIB should be compatible with schema as PIB has provisioning instances for considerable number of devices of the network. PIB is common to both PDP and PEP and is used to identify the type and purpose of policy information from PDP as well as PEP.

Simple network management protocol(SNMP) is an application level protocol that facilitates the exchange of management information between network devices. It is part of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. NMS execute applications that manage and monitor managed devices. SNMP is a simple request/response protocol. Management information base(MIB) is a hierarchal collection of information used by SNMP. MIB consists of managed objects, which represents any number of characteristics of managed devices and they are identified by object handles. In DiffServ scenario BB can be seen as NMS and edge routers as

managed devices.

Telnet is the most simple approach to configure routers. Routers that support this can be configured completely with one Telnet session. If this approach is used then BB should be able to start a Telnet session on the edge routers when they are to be reconfigured. Any scripting language can be used to automate this process when a policy decision is to be enforced at edge router.

3.2.3 Inter domain Communication Protocols

There are number of protocols that fulfill bandwidth broker's intra domain needs discussed in the section 4.2.2. However when it comes to inter domain level, there is no single protocol that fits into the requirement of BB. Due to this complication Internet2 QBone BB advisory council has proposed a simple inter domain bandwidth broker signalling protocol(SIBBS) in [31]. SIBBS is in its definition phase and only its basics have been finalised. There are alot of issues regarding SIBBS that have proposed solutions without any final answer.

Simple Inter domain Bandwidth Broker Signalling(SIBBS) follows request-response model between peer BBs and is sender oriented[31]. BBs have long running TCP connections with one another, TCP provides the basic reliability and flow control. SLAs are not negotiated by the BBs, human operator needs to negotiate as well as communicate these SLAs to the relevant BBs. Whenever a BB receives a Resource allocation request(RAR) it checks sender's authentication, the route, egress router for the flow, SLA related to user or flow and policies related to the flow. RAR can be from a user in the domain or from another domain's BB. Resource allocation answer(RAA) represents the response to any particular RAR containing the parameters depending upon the RAR as well as response i.e;success or failure.

The most basic scenario for SIBBS is when a user sends RAR to BB and BB checks the validity of request. If the RAR succeeds the check then BB sends RAR with its own ID to the next downstream domain's BB in the path of the flow. However, in the case of failure, RAA is sent back to the sender of RAR. The process continues until the destination domain's BB receives the RAR. It checks and sends it to the destination end host, which on success returns RAA to local BB. Then the backward transfer of RAA starts in a way that every intermediate BB firstly configures its routers and then sends success RAA to upstream BB. SIBBS also support tunnels, if one or both ends of the reservation are not fully specified then it forms a tunnel. This kind of request can be from end host or BB who has some kind of aggregation algorithm for the flows. In all transit domains except penultimate domain the behaviour of BB is same. In penultimate domain, in addition to performing routine checks the BB makes core tunnel voucher

and adds it to RAA, which is sent to the destination domain in case of success. In the destination domain after performing the checks local BB returns RAA with the voucher to penultimate domain's BB. The take down is automatic due to the presence of time stamp but for backup semi soft state mechanism is maintained.

Security in SIBBS is one of the main issues due to sensitive information transfer related to SLAs. Use of public keys for security in SIBBS is proposed. There can be two methods that can provide security, first is adding signature of the logical source in the messages transferred within SIBBS, second is mutual authentication.

4 Research Directions

Bandwidth Broker has been a very active research topic for last few years. In the following Subsections salient features and overview of current implementation status of some of the major efforts by different research groups are discussed.

4.1 QBone Bandwidth Broker Work Group

Among all the ongoing efforts of designing BB, the most significant role has been played by the Internet2 QBone Bandwidth Broker Advisory Council. It was one of the first group that has defined the BB's requirements[27] in detail and the initial draft of simple inter domain bandwidth broker signalling protocol[31]. The BB model and SIBBS model presented in this paper is based upon the Internet2 QBone Bandwidth Broker Advisory Council model. The current group (QBone Bandwidth Broker Work Group) is an evolution of the Bandwidth Broker Advisory Council that was begun in 1999. The current group has been working since 2000 to fill in the details of the SIBBS protocol.

4.2 CANARIE ANA

The objective of CANARIE ANA project is to implement a basic BB that is capable of providing differentiated services for CA*net II. CA*net II is a new high speed network for research and educational institutions that makes possible to run and manipulate advance applications, such as multi-media conferencing. The model to be followed is proposed in [12].

The basic functional modules of CANARIE ANA Bandwidth Broker as shown in Figure 3 are described below:

- The BB database provides application programming interface(API), to be used by broker. BB database stores all the information about SLAs and bandwidth allocation;

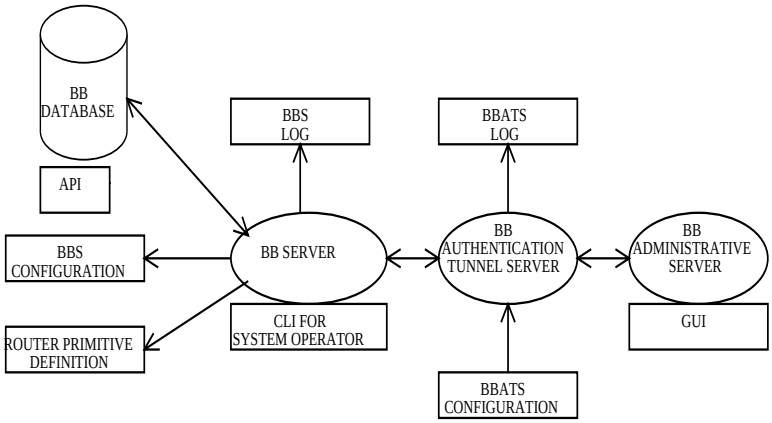

Figure 3: CANARIEANA Bandwidth Broker

- BB server is a multi process program implemented to operate within its trusted region. The main functionality provided by BB server is data validation, policy enforcement, update of transaction log and clean up of data. BB server and BB authentication server must reside on the same machine as BB authentication server provides the necessary security to BB server;
- BB authentication tunnel server provides necessary security to BB server;
- BB administrative server (BBAS) manages billing and contract management services and keeps customer's/client's information.

The BB server is configured by BBS configuration file which has information about port to listen on, where to store log file, services offered, router primitive definition and other information about routers in its region. Router primitive definition file contains commands to configure any specific router. BBS maintains flat text log file about all the transactions taken place.

BB transfer protocol(BBTP) is designed for client and BB interaction[11]. BBTP is based on client/server model and it is a request response protocol with client initiating the request. A client establishes a connection and generates request. The server responds to the client's request appropriately and closes the connection afterwards. BBTP communication generally takes place on TCP/IP connection. However any reliable protocol can be used. There are two types of messages for BBTP, request and response. The header types of BBTP messages are general header, request header,

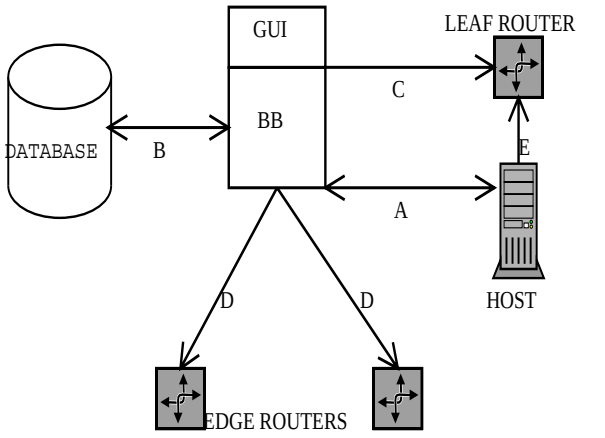

Figure 4: Architecture of Kansas BB

response header and entity header. Three classes of status code are defined for success, client error and server error respectively. When it comes to security considerations, BBTP does not provide any mechanism to authenticate clients.

This project has implemented the features such as: fixed service class definition, definition of local SLA, local static provisioning, distributed static provisioning, router definition facility, operator controlled router configuration, static report generation, secure broker interaction and transaction logging[1]. The implementation under this project can support the functions of storing SLA information, accepting resource request and configuring routers in a single domain only. Secure web browser based interface or a command line interface is used to control the BB. BB communicates with routers in its domain with telnet interface.

4.3 University Of Kansas

University of Kansas research group makes its bandwidth broker as in charge of internal as well as external affairs and its basic model is proposed in [2]. Internal in charge means that it keeps track of user's QoS requests and allocates resources considering domains policies. On external level BB provides QoS to its border crossing traffic by maintaining bilateral SLAs.

The functional model of BB is shown in figure 4. The database in-

cludes SLA, bandwidth allocation request and mapping between bandwidth to diffserv code point (DSCP). Bandwidth allocation request(BAR) saves information about SLA, leaf router, destination, source, rate and time. To provide desired level of service in its domain, BB keeps information about the role of every router in the domain as well as bandwidth and DiffServ capabilities of each router.

The client interacts with BB directly by bandwidth broker transfer protocol[11] (A in figure). BB consults the database for verification(B in figure). On verification BB sends the policing and marking parameters to the leaf router mentioned in the BAR(C in figure). BB configures the appropriate egress router by passing the essential parameters to the router(D in figure). Host then starts sending traffic through leaf router(E in figure).

BB model is implemented as client/server architecture. BB is the server that on start-up connects to the routers in the domain and performs their basic configuration. Either hosts in the domain or the network administrator can generate requests as client. A policy database in MySQL in maintained by BB server. An Apache web server is used to provide interface between an authorised user and BB. SLAs are added statically by the network administrator.

Resource allocation requests(RAR) are generated by the DiffServ host. On receiving the request BB checks the RAR against its SLA. If the request is accepted then BB sends back the success message after reconfiguring the edge router and DSCP is sent to the host to be set in the packets of the flow. In addition to admission control BB also performs validity check. When a flow expires its resource allocation is teared down. Moreover BB maintains a database of SLAs, when SLA expires all RAR entries related to that SLA are deleted. DS daemon running on each host estimates the required bandwidth by means of *setsocket* system call to the kernel. The same daemon is responsible for generating requests on behalf of host to BB. BB keeps a database about the routers of its domain. BB needs connectivity to edge routers for reconfiguration and the core routers are configured statically.

4.4 Merit

The research group at Merit[30] has proposed a multidomain bandwidth broker model in which with the help of some BB functionalities the support for virtual leased lines(VLL) can be implemented. The model due to its narrow scope only focuses on the role of BB in authorising and establishing one type of service i.e; VLL, which is actually expedited forwarding(EF) PHB in multidomain scenario. In this model there is support for two types of VLL services.

- Long term VLL are established for long period of time and their establishment does not require any kind of signalling. BB concerns to long

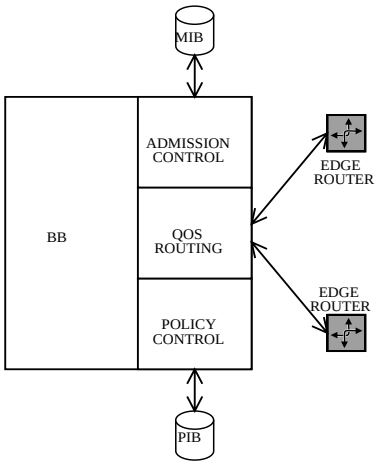

Figure 5: Functional Modules of Novel BB

term VLL only to the extent of managing the data flows and routing.

- Short term VLL requires explicit signalling for its establishment and termination.

If VLL extends to other domains then BB of local domain contacts other domains and negotiates the QoS parameters for VLL. There can be two types of SLS, committed SLS and open ended SLS. Committed SLS commits to support certain bandwidth. Open ended SLS does not have specified destination so has no commitment and reliability.

In transit domains the BB keeps track of two types of VLL. At intra domain level BB should be able to perform routing as well as admission control decisions. BB keeps complete knowledge of policy as well as network topology. In inter domain level BB responds according to RAR received form other domain's BB. There is no requirement of management for routers by BB in this case.

4.5 Novel

The main goal of bandwidth broker architecture developed by Novel research group is separating QoS control from core routers to provide much needed scalability in the network for guaranteed traffic [34]. The novel BB relies on virtual time reference system for QoS abstraction from the data plane.

The main functional modules of BB are shown in fig5.

- QoS routing module contacts the routers to get network topology to find the best available path for the particular flow.

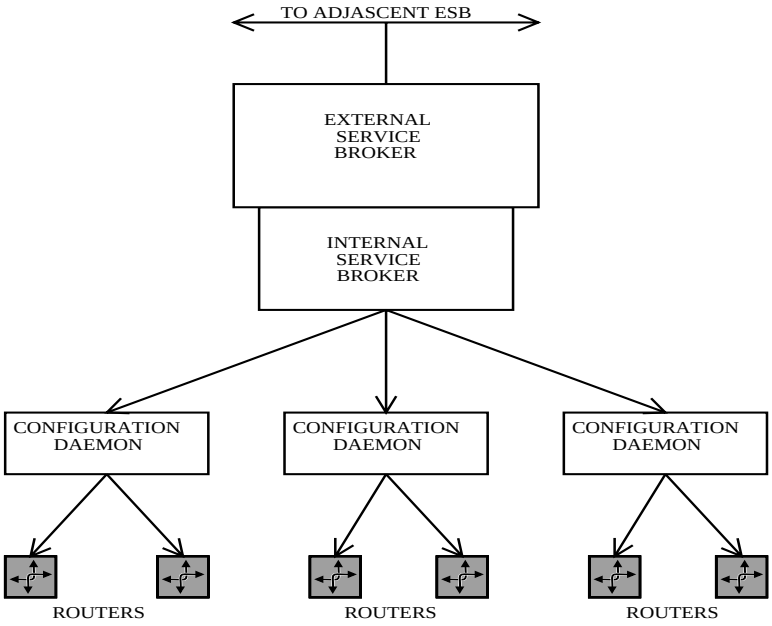

Figure 6: Service Brokers Hierarchal Model

- Policy control module keeps PIB for policy decisions.
- Admission control module performs admission control and resource reservation with the help of QoS states in MIB.

Novel BB provides class-based service guarantee with dynamic flow aggregation as well as per flow service guarantee within a domain. When a flow wants to enter BB's domain the ingress router of the flow requests BB for support of QoS parameters. After checking the policy and other information bases, if the request can be fulfilled BB selects appropriate path for the flow. BB sends success message to ingress router and updates its database.

4.6 Charging and Accounting Technology for the Internet(CATI)

The main aim of charging and accounting technology for the Internet (CATI) project is the implementation of charging and accounting mechanism based on the currently available IP protocols[8]. Secure interconnection between private networks was the reason of emergence of virtual private networks(VPN). QoS support can be introduced in VPNs with the help of service broker. Service broker sells the services according to specific terms. It has the ability to negotiate service cost with customers and setup the service on approval of agreement by both. The brokers are designed in a scalable hierarchal fashion[8] as shown in Figure 6.

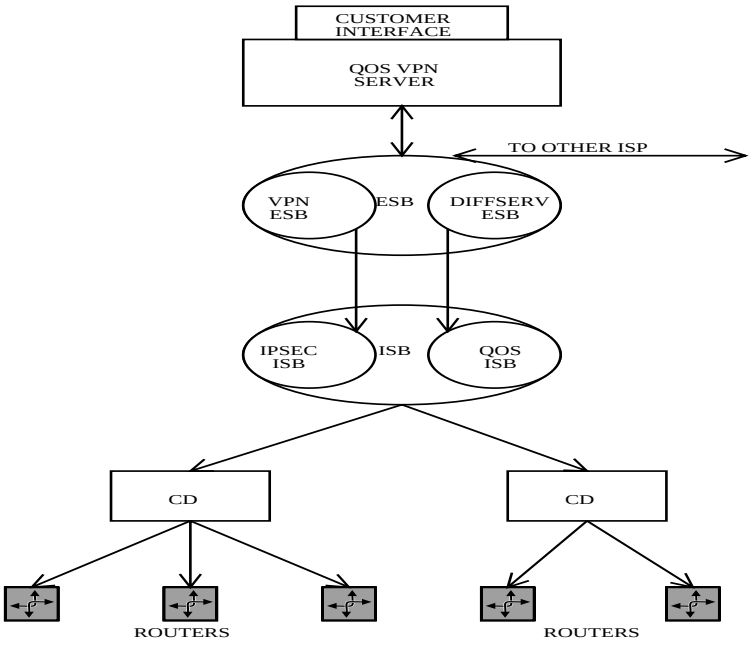

Figure 7: Interaction Among Service Brokers

The configuration daemon(CD) is at the bottom of hierarchy and is responsible for device level configuration. Internal service broker(ISB) is responsible for management of services in the domain level. External service broker(ESB) interacts with other domain's ESB for resource management. It can be viewed as BB at this level. The main aspect of this project is to implement effective charging and accounting functionality between ESBs. The electronic SLAs established between ESBs describe the charging method of services. When a customer requests a VPN connection with specific QoS from its ISP, the request is handled by QoS-VPN server. The QoS-VPN server contacts local ESB(VPN ESB and DiffServ ESB). ESB negotiates with next ESB and so forth until the request reaches the destination domain. In case the request is accepted by all ESBs, the respective ISBs on the route from source to destination configures their routers to support the requested QoS. However the ISBs for the border domains configure their routers for VPN tunnel. The interaction among service brokers is shown in figure 7.

Costumer negotiates the price as well as service required with its ISP via ESB[6]. ESB also checks with other ESBs in case the flow has destination in other domain. The process can lead to a chain of requests before the ESB is able to calculate the total final cost. If the costumer agrees with the cost, the service starts. There can be different charging mechanism like one time charging, continuous charging or usage based charging. The security of a component and security of communication between the components is

relative to the position of the component in the hierarchy. ESB is at the top most level in the hierarchy and has more intelligence and power over the network so it can cause more damage. Therefore it requires more secure communication methods. The security of the interactions is achieved with the help of encryption algorithm. The key length is calculated on the basis of severity of security threats.

The Implementation

The implementation of CATI project is under way and there are some issues that are left for future research[20]. For the time being, the implementation is only to the intra domain level where CD configures the routers and ISP makes the admission control decisions. A charging and accounting mechanism is also implemented which is like a small part of ESB. However the interaction between ESBs on interdomain level is not implemented yet. A user request to the ISB contains QoS parameters. ISB performs following three checks:

- First of all ISB checks SLA database for user verification;
- Secondly ISB checks connection database for existence of same connection;
- At the end if both checks succeed then the resources are checked in the resource database.

At the time of tear down ISB consults pricing database to calculate the price of the services provided.

4.7 Globus Architecture for Reservation and Allocation(GARA)

In providing end to end QoS the main problems are dynamic discovery of the resources and advance/immediate reservation of heterogeneous resources that are separately administered. GARA project tries to solve these problems and in the effort has built a prototype as part of the project[19].

GARA extends the idea developed in Globus resource management architecture. GARA provides management for separately administered resources. The architecture consists of three main components, an information service, local resource managers and co-allocation agents. An information service defines hierarchal name space and standard access methods to access the resources with the help of LDAP. Application that wishes to make reservation passes the request to co-allocation agent, the agent computes the resource requirements and directs the request to local resource manager or Globus resource allocation manager(GRAM). GRAM takes the request, authenticate it and if the request is successful then directs the local scheduler to allocate the resources and returns the job handle to the application.

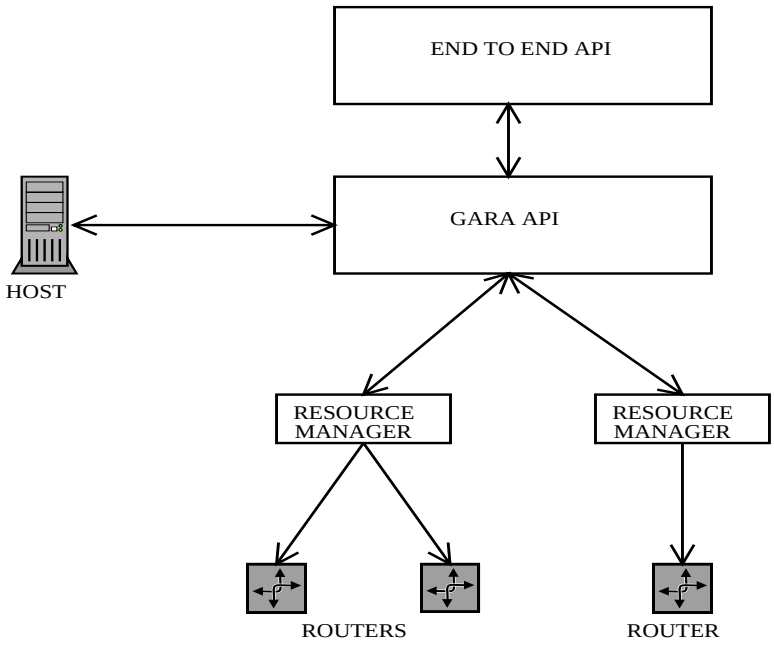

Figure 8: Architecture of GARA Implementation

GARA adds extra functionality for advance reservation of heterogeneous resources to the basic model. The support for heterogeneous resources is provided by introducing generic resource object, that encapsulates network flows, memory blocks, disk blocks and even processes. It gives generalised functionality in this way. The support for advance reservation is provided by separating the reservation from allocation. For immediate resource allocation request allocation is performed at the time of reservation, but in case of advance reservation only reservation handle is returned and the resources are to be reserved at the time of start of the service mentioned in the request. This advance functionality is provided by introducing a new entity i.e., co-reservation agent. Its function is similar to co-allocation agent except that after calculating the resource requirement for advance reservation, it does not allocate but simply reserves the resources.

GARA prototype provides in addition to some other features the functionality of bandwidth broker. No one entity in the GARA prototype is equivalent to BB but the whole prototype manages to provide some of the basic functionalities associated with BB. The working model implemented by GARA is shown in Figure8.

GARA provides an end-to-end API to support reservation between source and destination. Resource managers are like intra domain BBs. They are responsible for admission control and traffic shaping as well as traffic marking on intra domain level. The end-to-end API uses GARA API to contact

the resource managers to perform reservation. GARA API also allows the user to contact resource manager to specify the requirements for reservation. Public key authentication is used to control the access to the resource managers. Resource managers enforce SLAs and control one or small number of routers. In case of end-to-end interdomain reservation, application accesses end-to-end API and communicates its request to it. When the request is accepted all the resource managers reconfigure their connected routers to support the flow. The resource manager controls the routers with TCL script containing the login, password, and the interface to configure the routers.

4.8 Comparison

The Table1 provides a summarised overview of all the implementations discussed in this section. The table compares the implementations on the basis of the basic architectural components of BB discussed in section 3. Almost all the models discussed in the previous subsections describe intra domain level implementation specifications. The interdomain level specifications of most of the models are not finalised. Almost all the implementations summarised in the table does not support dynamic SLAs. Moreover tha databases that these implementations use are not fully implemented. In brief, there is no implementation that provides a complete functionality of BB. The development and implementation of a comprehensive inter domain BB communication protocol is a big hurdle in this process.

5 Distributed Bandwidth Broker

There is a great concern about the scalability issues regarding bandwidth broker. There is a rapid growth of QoS applications like VoIP and real time content delivery, which require dynamic QoS control and management. The ability of BB to handle large volumes of flows is debatable. The badly designed BB can itself become the bottleneck to allocate the network resources effectively even in the scenarios when the network itself is underutilised. To overcome the scalability problems of BB, a multiple BB architecture is proposed[33]. There is one central BB (cBB) and number of edge BBs (eBB) in the domain. The QoS states are represented at two levels, link and path. Link QoS state database maintains information regarding each link of the domain. The path QoS database keeps information about all the paths in the network, which is extracted from link QoS state database of the links of the path. The model uses path-oriented quota based approach[33]. To limit the flow requests accessing the link database the bandwidth is allocated in units of quota to different paths on demand. The central BB keeps link QoS state database and is responsible for quota allocation among the eBBs. Every edge BB has mutually exclusive subset of the path QoS state database

Architecture	CANARIEANA	Kansas	CATI	GARA
Data Interface				
Routing Table	No	No	No	No
Data Repository	Stores SLAs	Policy Database	SLA/ connection resources/ pricing database	Router config parameters
Application/user/network manager interface				
User Interface	Yes	TCP connection	web based GUI	GARA API
Network Manager Interface	Web based or Unix CLI	Web based GUI	Web based GUI	GARA API
Intra Domain Protocol				
Configuration of routers	Telnet Interface	Telnet for Cisco routers, TCP connection for LINUX routers	Configuration Daemon uses scripts	Resource manager
Admission control	Not specified	Not specified	Interior service broker	Performed by resource manager
Security	Not specified	Not specified	MD5 and SHA-1	Strong authentication
Inter domain Protocol				
Main Functional Module	Not specified	Not specified	Not specified	end-to-end API

Table 1: Comparison of Different Implementations

and perform admission control for the corresponding paths. When flow arrives at the edge router, the request to allocate resources to that flow is sent to the eBB that is responsible for that edge router. If the eBB has enough resources then flow is admitted. However if not sufficient bandwidth is available to meet QoS requirements of the flow then the request is sent to the cBB, which will perform admission control by checking available bandwidth of all the links along the path in link QoS database. The above architecture does provide a way to handle the scalability of BB but the architecture itself has some drawback. To reduce the processing overhead at the BB, the ability of the BB to perform effective and optimal resource management is degraded by introducing path oriented quota based bandwidth allocation mechanism. Increasing the size of quota can greatly reduces the effective allocation of bandwidth to flows. Multiple eBBs have allocated paths for which they perform admission control, but the distribution of bandwidth of links that are common to number of paths is ambiguous. The same aspect also introduces bandwidth wastage along these paths. If the link database is accessed frequently then increase in the processing overhead does not justify the selection of quota-based approach.

6 Pricing

Pricing and billing issues are very important in assessing the feasibility of development of DiffServ. DiffServ is the architecture that provides flexibility for service providers to introduce new and useful services, the whole effort wont be justified in terms of economical benefits if there is no flexible pricing model and billing architecture [21]. Moreover the pricing issues need special attention in DiffServ as the core routers cannot participate in the pricing model and only edge routers can perform required signalling needed for pricing [23].

Bandwidth broker is responsible for negotiating SLAs. Users pay according to agreed SLA and the provider has the responsibility to fulfill the SLA, otherwise provider can be penalised. BB is the only entity in the DiffServ network that works as a link between user and provider, hence BB is the best possible entity that can have pricing mechanism. In this section a pricing architecture which has BB as its main functional module is briefly described.

Dynamic Capacity Contracting [23] architecture implements bandwidth broker as a negotiating agent between users and service providers as BB has all information regarding SLAs and policies. BB provides the mechanism to user to choose the service provider with the least cost for the required service. The prices of services provided by service providers are congestion based in this scenario. It means that the price may vary with the variation in the network congestion.

BB enables users to negotiate the term contracts with service providers. Negotiations of term contract is a highly dynamic short-term process. The edge routers maintain a list of active contracts, their terms and their already offered prices. The edge routers update the list based on low granuality timer and delete expired contracts. These edge routers have the mechanism to calculate the congestion of a network. These router updates next to be advertised price table, for the future contracts based on the network condition. When user needs to send traffic, BB queries on behalf of user about the current advertised price. Dynamic capacity contracting provides the architecture to service providers to optionally offer value added services on top of flat rate services. User can dynamically choose any type of service for any time period.

7 Service Level Agreements

SLA specifies the forwarding services that a customer should receive and it may include traffic conditioning rules. BB must have a functionality to store, implement and update SLAs according to the customer's request as well as the network conditions. The introduction of some advance pricing functionality can increase BB's performance in terms of SLA negotiations. Due to the complex nature of SLA, advance research reveals some alternates to define SLA range. Some of these ideas are briefly discussed in this section.

7.1 SLA Traders

SLA trader is an agent that has an integrated mechanism for resource allocation, path selection and pricing into its trading process. The idea of SLA traders is proposed in [17]. BB can be transformed to a SLA trader by introducing the routing and pricing issues in the SLA. The most complex form of SLA trading is when the negotiation is among two ISPs. ISP can sell the resources that are directly owned by the network provider or the resources that are bought by this ISP from other network providers. The negotiation mechanism among SLA traders is one of the main functionality that they should support. The advantages of integrating SLA trading functions with BB mentioned in [17] are the following

1. Better efficiency by selecting least cost links;
2. Simplified configuration of DiffServ boundary and SLA setup;
3. Deployment of routing, pricing and provisioning policies locally;
4. Optimised resource allocation for aggregated flows[17].

Initially SLA traders have the ability to make local decisions such as what resources are to be given to which peer. Once there are offers from

different SLA traders about the services that they provide, then those offers can be accepted or rejected. Accepted offers can result in evolution of new services. The cost of these new services is sum of SLA price offered by peer plus the cost of ISPs own resources. Whenever there is a formation of new service the SLA trader compares all the offers from the peer SLA traders and choose the best one according to the quality, price and policy. SLA traders store established contracts in the database. They have the ability to performs routers reconfiguration when required.

For the communication among SLA traders a very simple protocol SLA trading protocol(SLATP) is proposed in [16]. SLATP can be implemented on top of any datagram service due to its simplicity. SLATP owes its simplicity to the fact that SLA traders need to communicate with their peers only. The *bid* message is sent by SLA trader to advertise all its available resources to its peers. It can be answered by *accept* or *reject* message based on the decision by the peers. In case of acceptance, *confirm* message is sent to seal the contract. However, there is a special message type, *ask* message which is used to ask for the bidding from peers, when the speeding up of the process of SLA trading is required. The pricing strategy used in SLA traders is residual bandwidth pricing, in which the price gets higher as resources get lesser.

There are two algorithms to determine the need of resources, the passive provisioning algorithm and active provisioning algorithm. Passive algorithm waits for the request from the customer before calculation of resources but the active algorithm tries to forecast the future needs of the resources. When SLA trader wants to buy a bid it evaluates the worth of bid by profitability analysis algorithm, to check whether money can be saved when the bought services are to be sold later.

7.2 Range based SLAs

The range based SLA specify the lower and upper limit of the traffic that can be transferred, instead of one fixed value normally specified in SLAs [10][9]. The range based SLA's concept was proposed during the development of CATI project discussed in section 4.6. Range based SLA helps in reducing the waste of resources due to user's inability to specify the exact resource requirement[10][7]. This section has a brief description of a mechanism to implement range based SLAs with the help of BB.

7.2.1 Virtual Core Provisioning

Virtual core provisioning in BB architecture provides a mechanism to implement range based SLAs where ISP guarantees the lower bound of bandwidth only [10]. However, the architecture also supports fixed value SLAs for user's convenience. The model proposes that the edge device should be responsi-

bile for provisioning and the core devices require no explicit configuration. The advance reservation states at the core are maintained in the capacity inventory of BB system [7]. In the proposed architecture there are some assumptions. First of all the direction that the traffic follows after entering at ingress router is known. Secondly the topology of network is known by BB so the traffic can be forced to pass through some specific nodes governed by MPLS and route pinning in the core network[10]. A logical partitioning of the bandwidth takes place at edge and interior routers. Traffic with fixed bandwidth reqirements are separated from those for which only upper and lower limit is known.

A simple algorithm is proposed that can determine the resources currently reserved for each type of traffic. Whenever there is a request of bandwidth, the edge router selects MPLS enabled pinned path. The resources are checked at the edge routers as well as at the core routers along the whole path. It is checked that the capacity at related interfaces of the core routers does not exceed the upper bound after acceptance of connection in case range based traffic. If the check results are positive the connection is submitted. The acceptance of the connection triggers the actual configuration of the edge routers however only the resource states of core routers stored in BB is to be updated.

7.2.2 Dynamic Edge Provisioning

Dynamic edge provisioning model is a two layer model, where upper layer controls the lower layer. Upper layer controls the edge provisioning and lower layer is responsible of interior resource provisioning. The idea is proposed in [9]. Bandwidth broker is responsible for end to end admission control as well as managing and provisioning network resources. BB has the additional functionality to divide the capacity at edge routers into different groups and manage them efficiently to allow resource sharing among the groups in dynamic and fair manner.

At edges and interior routers is dedicated for VPN connections. At edge routers this bandwidth is logically divided among the dedicated VPN tunnels and those connections that are willing to specify rate by the range. Dedicated VPN tunnels are the connections with one fixed value of reservation of resources. The bandwidth reserved for range based connections is further divided among different groups of VPN connections where each group supports a different range of bandwidth. At the edge the reserved bandwidth for VPNs is rate controlled by policing and shaping however at interior routers this capacity indicates the bandwidth that can be allocated to quantitative traffic in case of requirement.

There are number of policies that can be implemented to facilitate the sharing of bandwidth among groups. The base capacity is shared among all the groups so they are called shared service groups. The connections that

does not support range based SLAs are grouped under dedicated service group. There are three main categories of policy that can be implemented to facilitate sharing among groups.

1. First policy does not allow any sharing even among own shared service group. It means that capacity unused by one group cannot be used by any other shared service group even when that shared service group needs it. The implementation of this policy is very simple, at the time of admission only the availability of minimum bandwidth in the shared service group is checked by BB.

2. The second policy allows the shared service groups to borrow from each other however the shared service groups cannot borrow from dedicated service group, however the borrowed bandwidth must be returned when the shared service group from where it is borrowed needs it. When there is a request for connection then in addition of checking the available capacity in the shared service group, BB has to take into consideration the capacity that is borrowed from this group.

3. The third and the most complicated policy to implement is when capacity is allowed to be shared among all groups, dedicated as well as shared service. This situation is quite difficult and complicated to manage as the unused bandwidth from dedicated group must be distributed fairly among all shared groups. There are few options presented in the architecture like

   - Allocation of unused resources to lower user groups first
   - Allocation of unused resources based on proportional needs
   - Allocation of unused resources to highest needy group first.

8 Active resource management

DiffServ and BB provide an architecture that enables the users to reserve the bandwidth to guarantee the required quality of service. There is a possibility that user over estimates his/her requirement due to lack of knowledge which leads to inefficient resource management. To incorporate this suboptimal usage of network a mechanism, active resource management(ARM) is proposed in [29]. ARM uses monitoring service to monitor the flow and reallocates the unused bandwidth without loosing the quality of service. All the traffic is marked with some DSCP which actually reflects the SLS for that flow. To implement ARM, BB monitors the traffic rate with the help of monitoring meters and when it monitors that the traffic rate is less than the allocated one then it knows that network resources are being wasted. The unused bandwidth is returned to the unused bandwidth pool. There is no

reserve bandwidth for best effort services, BE can get bandwidth from the unused bandwidth pool. If there is some time when all the users are sending at their peak rate, the bandwidth will be allocated back to them from the unused pool or from the best effort pool.

9 BB in Different Network Architectures

Bandwidth broker is introduced in[28] to be implemented in DiffServ domain for controlled sharing of organisation's Internet bandwidth. With the advancement of research in this field, it is apparent that BB can fulfill some other requirements in other network architectures too. The following subsections have some very interesting discussions about incorporating BB in different network architectures to optimise the network resource management. Integration of BB in MPLS network for traffic engineering is briefly discussed. IntServ over DiffServ model is also discussed to briefly elaborate BB's role in providing QoS in that architecture. BB's importance in providing QoS and resource management in DiffServ and mobile network is briefly described in one of the subsections.

9.1 BB and MPLS

MPLS is one of the emerging technologies and seems to have number of useful features to provide QoS. BB can be introduced in MPLS architecture to solve some of its QoS issues.

9.1.1 RATES

Traffic engineering is one of the most significant reason for MPLS network deployment. Routing and traffic engineering server(RATES) is proposed for this purpose in [4] and it has many similarities with intra domain BB.

On reception of request RATES tries to find a new LSP without changing the existing ones. RATES provides a mechanism for setting up LSPs with bandwidth guarantee. However, SLAs can have other QoS parameters such as delay, jitter and losses. To get link state information RATES is implemented as link state peer in the network. A request triggers the computation of route for a new demand and request can come from either the ingress router or via GUI. RATES has complete information about network topology and provides recovery in case of link failure in shape of rerouting the traffic. RATES maintains a relational database for policies and provides GUI to this database for network administrator. Once RATES computes a route for an LSP demand, it communicates with the ingress router with COPS to setup the LSP. Moreover it sends policy decisions and reconfigures the routers according to the packet classification parameters. RATES scales

well within a single OSPF area. There are number of modules combined together to form RATES. COBRA based bus is used for communication among modules. GUI is provided for necessary management access for provisioning and monitoring. On receiving a request the routing module computes the route for the traffic by considering policies, network condition and request parameters. Upon link failure RATES performs restoration with the help of restoration-capable online routing algorithm. RATES includes a COPS policy server or policy decision point(PDP). PDP is responsible for reconfiguring edge routers for new LSPs.

9.1.2 Unified Layer 3 QoS approach

The basic model of the unified layer 3 QoS approach is derived from DiffServ. The model is introduced in [14]. The main functionality provided by different components of the architecture is discussed below:

- QoS manager/bandwidth broker performs admission control, manages network resources and pricing;
- Domain hosts have applications that can specify QoS requests to QoS manager/BB;
- Leaf routers perform per flow policing and shaping;
- Core routers does not have any advance functionality as they are suppose to perform fast switching;
- Domain edge routers are responsible for inter domain traffic policing and shaping.

The model supports future reservation based on flexible QoS negotiations. If the request cannot be fulfilled due to the lack of resources then alternate choices can be given to user in the form of degraded QoS which can be supported then or in near future. However in case of unavailability the busy signal is passed to the user. It is application designers responsibility to map QoS parameters to bandwidth requirement, delay tolerances and reliability which can be understood by BB.

Once the request is accepted then BB configures each router on the predetermine path of flow for policing. Leaf ingress router performs initial labelling of datagrams which can be translated for each datagram at the core routers to perform policing on aggregated flows. At egress router the label is stripped off and datagram is forwarded to the destination. The basic concept is to enforce fast MPLS label based forwarding of datagrams in DiffServ network where routing is based on translation of tags. There is a requirement of label distribution protocol which is fulfilled by distributing labels by BB with QoS configuration parameters.

9.2 BB in IntServ over DiffServ model

The IntServ architecture has scalability problem whereas the DiffServ architecture does not provide very strong guarantee of service. The best of both the architectures can be captured by merging them together in such a way that weaknesses of one overcomes by strong points of other architecture. The best possible network architecture is achieved by integrating IntServ with DiffServ.

9.2.1 Basic Model

Implementing IntServ in the stub networks fulfill the users requirement of strong guarantee for the services agreed in the SLAs. Users make requests using RSVP in the stub networks. The transit domains are DiffServ capable, and flows are aggregated on the basis of DSCP. DiffServ in the transit domain solves the scalability problem and the routers of transit domain do not need to maintain per flow information.

The basic model of hybrid network consisting of IntServ networks as stub network as well as a clients of a DiffServ transit domain that connects the stub networks, is introduced in [6]. The signalling protocol in IntServ network is RSVP. The hosts are assumed to use RSVP signalling to request QoS level as well as some of them have ability to mark DSCP to their traffic. The edge routers have the functionality for RSVP and communicates with BB in IntServ domain. Moreover these edge routers are also DiffServ capable. The border routers of DiffServ domain are capable of providing traffic conditioning functions according to the SLAs negotiated between the two domains and also communicate with BB to perform traffic conditioning as well as resource management. Stub networks has IntServ capable hosts but it is not necessary that all routers in that network are to be IntServ capable in the later case they acts as non-RSVP cloud. Stub networks may use BB for providing QoS to their users. The transit domain is DiffServ capable and RSVP message can pass through this domain transparently.

IntServ networks classify traffic on the basis of flow spec and DiffServ networks classify it on the basis of DSCP. There is a need of mapping between these two classification mechanism. The most simple and straight forward default mapping can be to map controlled load service to AF and guaranteed service to EF. However there can be a mechanism that provisions for customer specific mapping. The edge router in the stub network process the host RSVP message then contacts the local BB of the domain. The local BB aggregates the flow requirements and request to the BB of the transit domain. The transit domain BB has to check its policy database and contact all other downstream transit domain BBs in the path of the flow if required. If the request is accepted by all transit domain BBs as well as BB of destination stub domain then all transit domain BBs setup appropriate

traffic profile in their border routers. The last downstream transit BB has to convey messages to the local BB of the destination stub network to reserve resources for the aggregated flow in its domain.

9.2.2 Admission control in IntServ over DiffServ model

A mechanism is proposed for admission control in IntServ over DiffServ architecture using BB in [22]. Every DiffServ class has assigned a priority value for a particular ingress/egress router based on the bandwidth allocation to that class to the links joining the ingress/egress pair. When RSVP flow requests the DiffServ domain for network services then the flow is assigned a PHB. On the basis of that PHB and the requested QoS parameters a token value is generated for that paricular flow. The BB has functionality to manage aggregated token value for link between all ingress/egress pairs. BB has the responsibilities of priority scheduling and providing guaranteed QoS. Every router stores class priority values and priority tokens for flows going through it. BB adds these values and take admission control decisions on the basis of these values.

9.3 BB in DiffServ and mobile network

The implementation of DiffServ architecture enables the mobile network to support large number of multimedia services as well as applications[26]. However, integration between these two type of networks is a non trivial issue and can open lots of areas of research. Few of these issues can be solved by introducing BB in the architecture. First and the main problem that can be faced is need of dynamic SLAs due to the ever changing location of mobile host. BB can provide support for this performance. The route optimisation approach encourages the mobile host to communicate with BB directly in the current network, a signalling protocol for this communication is required. When a mobile user moves from one domain to another then the BBs of the domains(home and foreign) should establish signalling communication with each other to exchange QoS information directly to avoid any black hole[26].

9.3.1 DiffServ and GPRS network

The compatibility issues among DiffServ and GPRS are discussed in [24]. Mobile communication is one of the most recent and ever growing technology. In addition of transferring voice, it is now being used for transferring data too. General packet radio service(GPRS) is one of the modern trends in mobile environment. In this section generally the working of DiffServ networks with GPRS is discusses and specifically the role of BB in this scenario is elaborated. Mobile station(MS) is the equipment intended to access the telecommunication services. Supporting GPRS support node(SGSN) is the physical entity that is responsible for communication between GPRS

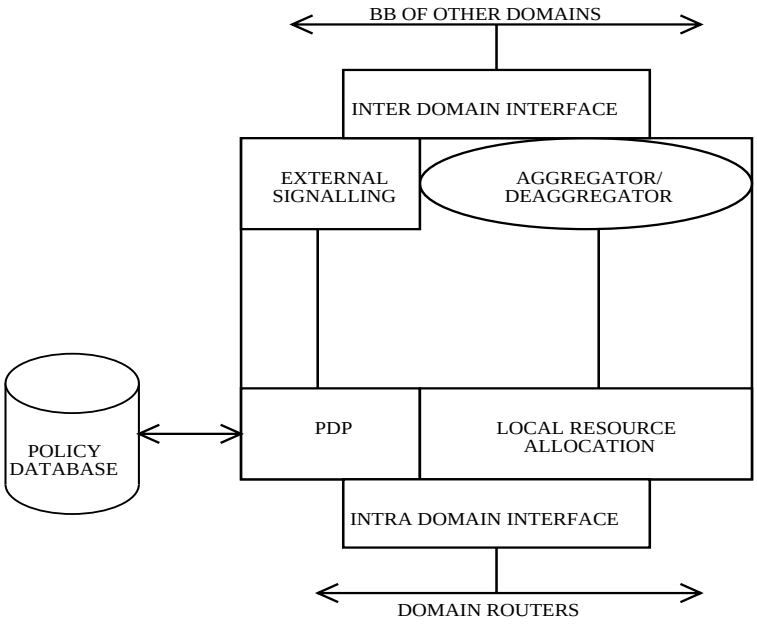

Figure 9: Bandwidth Broker in Access Network

network and all GPRS users in the service area. GGSN is the gateway of mobile network towards external networks. The functionality of GGSN is to be enhanced, in order to make it DiffServ compatible[24].

The architecture of BB in access network as shown in Figure9 emphasises the following:

- BB should be able to aggregate RSVP reservation messages in uplink direction and also able to deaggregate RSVP reservation messages in downlink direction;
- BB should be able to communicate with BBs of other DiffServ domains;
- BB should also have functionality for PDP, to configure PEPs so it maintains repository of the policies.

GPRS is a non DiffServ network so GGSN is used to connect GPRS with DiffServ domain by implementing a mechanism to mark the traffic before it reaches DiffServ domain. The BB of DiffServ domain sees GGSN as BB of GPRS domain as it has limited BB functionality.

Two different scenarios are discussed here. The first scenario is when remote host(RH) is in access DiffServ domain and GPRS domain is communicating to that RH through DiffServ capable network. In this case we can

see that GGSN is working as an aggregator and access BB is a deaggregator. MS sends its request to GGSN, which sends the request to access BB. Using COPS, access BB sends its approval decision to RH that it can use certain resources to access MS. Access BB also notifies the neighbouring BB about its decision. The neighbouring BB reconfigures its routers and sends the decision to GGSN. GGSN updates its path state and sends confirm message to access BB through neighbouring BB again. When RH sends its approval to access BB then the message is sent to SGSN through GGSN, SGSN then notifies the MS and the communication starts between MS and RH. In the second scenario, RH wants to initiate the communication with the MS so it sends COPS request message to access BB. Access BB translates it into RSVP end to end path message and tunnels it to GGSN with QoS profile. The GGSN, a deaggregator receives the message and translates it into "create PDP context request" message and sends it to SGSN. SGSN sends the message to MS, MS replies its approval with activate PDP context accept message. SGSN creates PDP response message and sends it to GGSN. The RSVP end-to-end reserve message is tunnelled to access BB, which sends COPS decision message to RH, which can send its approval by COPS reply message.

10 Conclusion

Bandwidth broker can perform admission control by managing policies and has ability to enforce these policies by configuring edge routers at intra domain level. On inter domain level BB enforces and manages SLAs dynamically. On network level it is evident that BB plays major role in providing end-to-end QoS to the users of its domain. Due to huge number of responsibilities handled by BB, a lot of research is needed in this area before a comprehensive BB can be standardised.

SIBBS is the only inter domain protocol defined partially for BB-to-BB communication. A complete framework of secure and efficient BB inter domain communication protocol is yet to be finalised. Security is one of the main issues in the development of inter domain communication protocol for BB. In addition to message integrity, end-to-end security of the communication is critical.

It cannot be safely assumed that BB has secure and trusted link to all the routers in the domain, hence the need of monitoring increases for the secure intra domain communication. Monitoring should be optimally integrated in the functionality of BB, to overcome the challenges due to the change of network topology.

SLAs are negotiated dynamically among BBs due to the ever changing nature of network traffic. There is a need to develop a mechanism by which BB can optimally support dynamic SLAs. The dynamic SLAs play a major

role in optimising the utilisation of network resources.

11 Acknowledgements

Authors acknowledge the support from Avaya Aus Labs, Sydney for this project.

References

[1] CANARIEANA bandwidth broker,final report 1. at www.gait .bcit.ca /projects/.

[2] Overview of bandwidth broker system. Technical report.

[3] Resource allocation protocol (rap), 2000.

[4] P. Aukia, M. Kodialam, V. Koppol, T. Lakshman, T. Sarin, and B. sutter. RATES:A server for MPLS traffic engineering. *IEEE Network Magazine pp.34–41*, Mar/Apr 2000.

[5] Y. Bernet et al. Differentiated services quality of service policy information base. Internet draft, IETF, Jun 1999.

[6] T. Braun, M. Gunter, and I. Khalil. Virtual private network architecture. Technical Report IAM-99-001, CATI, Apr 1999.

[7] T. Braun, M. Gunter, I Khalil, R. Balmer, and F. Baumgartner. Virtual private network and quality of service management implementation. Technical Report IAM-99-003, CATI, Jul 1999.

[8] T. Braun, M. Gunter, B. Stiller, and B. Plattner. The CATI project:Charging and accounting technology for the Internet. In *Multimedia Applications, Services and Techniques ECMAST'99, LNCS 1629*, pages 281–296, May 1999.

[9] T Braun and I. Khalil. Edge provisioning and fairness in VPN-DiffServ networks . In *The 9th International Conference on Computer Communication and network (ICCCN 2000)*, pages 424–433, Oct 2000.

[10] T Braun and I. Khalil. A range based SLA and edge driven virtual core provisioning in DiffServ-VPNs. In *The 26th Annual IEEE Conference on Local Computer Networks (LCN'2001), Tampa, USA*, Nov 2001.

[11] British Columbia institute of technology Canada. CA*netII differentiated services bandwidth broker transfer protocol. Technical Report 0.3, Nov 1998.

[12] British Columbia Institute of Technology Burnaby Canada. CA*netII differentiated service bandwidth broker high level design. Technical Report 0.4, Nov 1998.

[13] K Chan et al. Cops usage of policy provisioning (COPS-PR). Internet request for comments RFC3084, IETF, Mar 2001.

[14] Peter Clayton and Austin Poulton. Internet quality of service. In *1st South African Telecommunications, Networks and Applications Conference (SATNAC'98), University of Cape Town, South Africa*, Sep 1998.

[15] D Durham et al. The COPS(common open policy service) protocol. Internet request for comments RFC2748, IETF, Jan 2000.

[16] George Fankhauser and Bernhard Plattner. DiffServ bandwidth brokers as mini markets. In *Workshop on Internet Service Quality Economics, MIT, Cambridge, MA*, Dec 1999.

[17] George Fankhauser, David Schweikert, and Bernhard Plattner. Service level agreement trading for the differentiated services architecture. Technical Report Report No. 59, TIK, Jan 1999.

[18] M Fine et al. An architecture for differentiated services. Internet request for comments RFC2475, IETF, Dec 1998.

[19] I. Foster, Carl Kesselman, Craig Lee, Bob Lindell, Klara Nahrstedt, and Alain Roy. A distributed resource management architecture that supports advance reservation and co-allocation. In *Intl Workshop on Quality of Service, IWQoS '99 UCL, London*, Jun 1999.

[20] Noria Foukia and David Billard. Final Metrics for CATI. Technical Report 0.3, Apr 2000.

[21] Felix Hartanto and George Carle. Policy-based billing architecture for Internet differentiated services . In *IFIP fifth international conferenceon broadband communicatio,Hong Kong*, Nov 1999.

[22] Junseok Hwang and Martin Weiss. On the economics of interconnection among hybrid QoS networks in the next generation Internet. In *XIII Biennial Conference of the International Telecommucations Society (ITS) , Buenos Aires*, Jul 2000.

[23] Shivkumar Kalyanaraman, T Ravichandran, and R Norsworthy. Dynamic capacity contracting:A framework for pricing the differentiated services in Internet. In *Proceedings of 10th Annual Workshop on Information Technologies and Systems (WITS), Australia*, 2000.

[24] Georgios Karagiannis. QoS in GPRS. open report 5/0362-FCP NB 102 88 Uen, Ericsson, Dec 2000.

[25] G Kim, P Mouchtaris, S Samtani, R Talpade, and L Wong. QoS provisioning for VoIP in bandwidth broker architecture: A simulation approach. In *Communication networks and distributed systems modeling and simulation conference (CNDS)'01, Phoenix, Arizona, USA*, Jan 2001.

[26] Mikael Liljebladh and Ralitza Gateva. Integration problems between IP multicast and DiffServ concepts in mobile networks. Master's thesis, Umea University and Chalmers university of technology, Dec 1999.

[27] Rob Neilson, Jeff Wheeler, Francis Reichmeyer, and Susan Hsres. A disscussion of bandwidth broker requirements for Internet2 Qbone deployment. at http://www.merit.edu/i2qbone-bb/doc, 1999.

[28] K. Nichols, V. Jacobson, and L. Zhang. A two-bit differentiated services architecture for the Internet. Internet request for comments RFC2638, IETF, Jul 1998.

[29] Ananthanarayanan Ramanathan and M Parashar. Active resource management for the differentiated services environment. In *Third annual international workshop on active middleware services in conjunction with the Tenth IEEE International Symposium on High Performance Distributed Computing (HPDC-10) San Francisco*, Aug 2001.

[30] D Spence. Multidomain bandwidth broker model. www.internet2.edu /qos /qbone/info /bb-model.doc, 1999.

[31] B. Teitelbaum and P. Chimento. Qbone bandwidth broker architecture. Work in Progress. at http://qbone.ctit.utwente.nl/deliverables/1999/d2/bboutline2.htm, 1999.

[32] B. Teitelbaum and R. Geib. Internet2 QBone: A Test Bed for differentiated service. In *INET'99, The Internet Global Summit, San Jose, CA, USA*, Jun 1999.

[33] Z Zhang, Z Duan, and Y Hou. On scalable design of bandwidth brokers. In *IEICE Transaction on Communications,E84-B (8)*, Aug 2001.

[34] Zhi-Li Zhang, Zhenhai Daun, Yiwei Thomas Hou, and Lixin Gao. Decoupling QoS control from core routers:A Novel bandwidth broker architecture for scalabale support of guarenteed services. Technical Report 00-028, University of Minnesota, Mar 2000.