Paper

Bandwidth broker extension for optimal resource management

Shaleeza Sohail and Sanjay Jha

Abstract — Bandwidth broker (BB), resource manager of differentiated services domain cannot provide per domain behavior (PDB) attribute information to customers and neighboring domains at the time of service level agreement (SLA) negotiation. Extending BB's functionality to calculate PDB attributes can help it to negotiate SLAs dynamically and efficiently. Using current measurements or historic data about PDB attributes, bandwidth broker can perform off-line analysis to evaluate the range of quality of service (QoS) parameters that its domain can offer. Using these values BB can perform optimal capacity planning of the links and provide better QoS guarantees.

Keywords — bandwidth broker, per domain behavior, resource management.

1. Introduction

In order to support quality of service in the network, new architectures 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 domains by enforcing the aggregate traffic contracts between domains [2]. At the interdomain boundaries, service level agreements specify the transit service to be given to each aggregate [11]. 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 per hop behavior (PHB) to be applied to each aggregate. PHB is the treatment that a packet receives in a DiffServ domain at any router. All traffic belonging to a particular class experiences same PHB. To automate the process of SLS negotiation, admission control and configuration of network devices correctly and to support the provisioned QoS, each DiffServ network may be added with a new component called a bandwidth broker [13].

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. Organizational policies can be configured by using the mechanism provided by BB. On the inter domain level BB is responsible for negotiating QoS parameters and setting up bilateral agreements with neighboring 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 [6], 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 utilization.

Currently BB keeps no information about values of QoS parameters that it can offer. Some time critical applications or their users may need to know the exact treatment that their application will get in terms of delay, jitter, packet loss etc. For example in case of multi-party tele-conferencing, a user may need guarantee that his/her application's traffic will not suffer end-to-end delay more than 50 ms. The Internet service provider (ISP), using DiffServ in its domain can only guarantee that the user's traffic will be assigned to a particular behavior aggregate (BA) and PHB. ISP can guarantee the PHB that the aggregate traffic will experience but cannot guarantee the QoS parameters like delay, jitter and packet loss etc. To know these attributes ISP needs to know the per domain behavior of the domain. PDB is the edge-to-edge treatment that traffic receives in a DiffServ domain [7]. In order to efficiently negotiate SLS in this scenario and satisfy user's demands an ISP can use BB to calculate these QoS parameters for different classes of traffic. BB can perform off-line analysis on the current results or historic data and find out the QoS values that it can offer. In this manner BB will have a complete knowledge about the range of QoS parameters supported by the domain at any particular load condition. In order to improve the QoS values BB can negotiate SLAs dynamically with the neighboring domains.

The rest of the paper is organized as follows: Section 2 has a brief description of BB. Section 3 describes per domain behavior and related works are mentioned in Section 4. Section 5 relates BB with PDB. Section 6 reports on the simulation studies that we performed. Section 7 elaborates the impact of calculating PDBs at BB. Section 8 concludes the paper and give some ideas for future work.

2. Bandwidth broker

The main resource management entity in DiffServ domain is a BB. The BB maintains policies, and negotiates SLAs with customers and neighboring 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 Fig. 1. The figure shows that when a flow needs

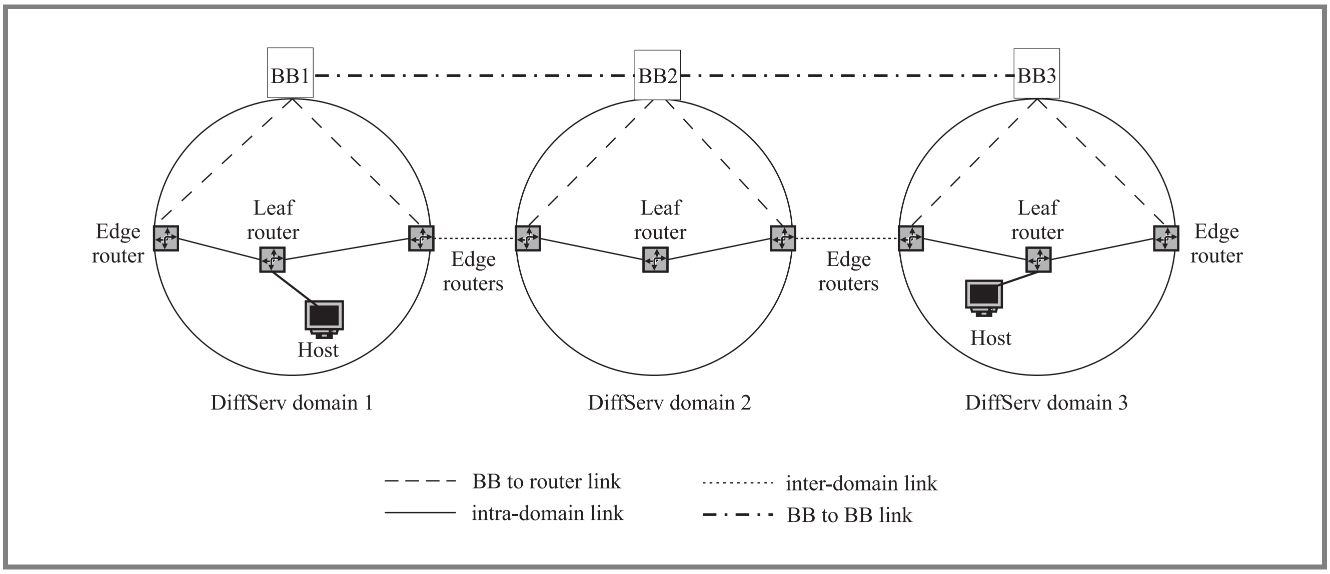


Fig. 1. Role of bandwidth broker in DiffServ.

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, a BB might have to negotiate a new SLA with the neighboring domain depending upon the traffic requirements. Once BB allows the traffic, the edge router or the 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. Per domain behavior

PDB consists of measurable attributes that define the treatment that each PHB will experience from edge-to-edge in a particular domain [7]. For example the PDB may specify the edge-to-edge delay that the traffic belonging to assured forwarding (AF) class may experience in the domain. PDB depends upon the PHB as well as the load conditions and some domain specific parameters like domain topology, links used to transfer traffic etc. The sum of same type of PDB parameters of all the domains from which the flow will pass gives the end-to-end QoS parameters for the particular flow. The attributes that can be part of the PDB are like delay, packet loss and throughput etc. The network specific parameters need to be specified for the measurement of these attributes [7].

4. Related work

The basic BB model is extended in virtual private network (VPN), supported by DiffServ to implement and negotiate range based SLAs [16, 17]. The resource wastage is reduced by using range based SLAs as the mechanism provides better resource utilization when user is unable to specify the exact resource requirement [15, 17]. IETF has defined PDB and the rules for its specification [7]. Multiple types of PDB are also defined; assured rate [9], virtual wire [5] and lower effort [8] are some of the examples. However ISPs can define their own PDBs according to their network requirements. Different research groups are studying the QoS attributes relation with the network parameters [1, 6].

5. Bandwidth broker calculating PDB

The bandwidth broker is a management entity that has a complete and up-to-date picture of the topology of the domain. Hence, the BB is the best possible entity that can be extended to calculate PDBs. In general the areas about which BB maintains information are policy, SLA, network management, and current resource allocation status [12]. Adding the functionality in the BB to calculate PDB and advertise them at the time of SLA negotiation can result in better user satisfaction. Moreover by knowing the PDB experienced by different PHBs, the BB can efficiently and optimally allocate resources.

The BB may choose to define a range of the QoS attributes supported by its domain by calculating maximum, minimum and average values of these attributes at various load conditions. BB can use these values to indicate the QoS treatment that any traffic may receive. To support particular value of QoS parameter BB uses this information for admission control as well as for SLA negotiation. For example BB may need to provide 50–100 ms of delay to any particular PHB. However from previously performed analysis BB knows that it is not possible at the present load conditions of the network. The solution is to negotiate the increase of bandwidth with the neighboring domains and considering the QoS requirement before accepting new

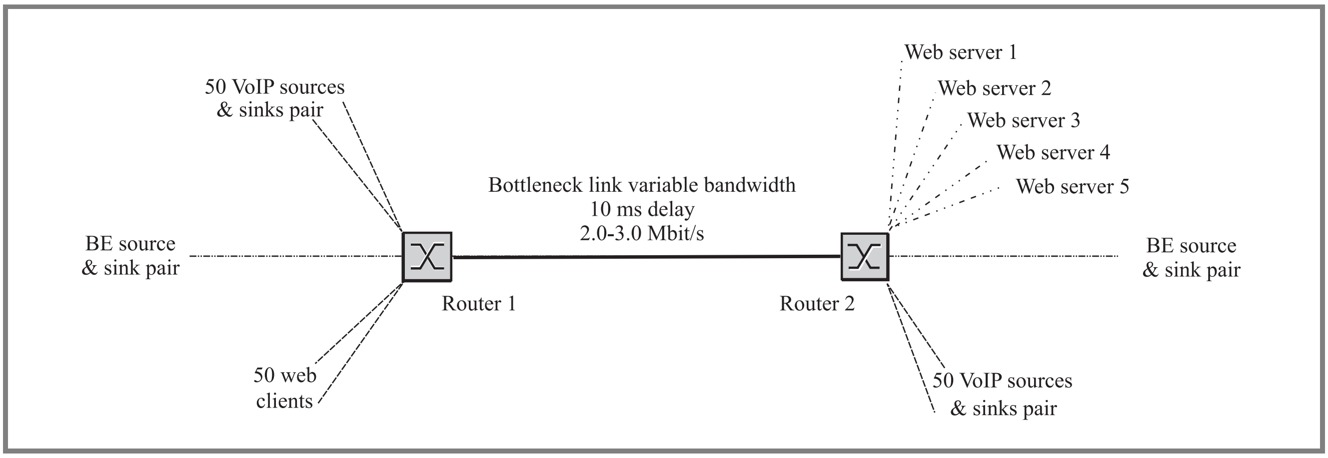


Fig. 2. Topology. All access links have 10 Mbit/s bandwidth and 0.1 ms delay.

connections. In this manner BB can optimally perform capacity planning of the links of the domain.

The simulation study in the next section calculates different values for some QoS attributes by changing few parameters. This simulation study shows that by using simple mechanism a BB can be extended to monitor different attributes of PDB.

6. Simulations

The simulations are performed using the network simulator (NS) [10]. Some of the simulation parameters are taken from the simulation study of DiffServ [1], however the scheduler used is weighted fair queuing [14]. In the simulation, the sources are generating traffic at constant rate and the bandwidth of the link changes for each simulation run. The values of QoS parameters change with the change of link capacity and the minimum link capacity can be found in this way that can support some particular QoS value. The impact of capacity on the attributes can help BB to decide what link capacity to use to transfer traffic, if certain QoS requirements like delay, packet loss etc; at a particular load condition, are to be fulfilled.

6.1. Simulation topology and parameters

The network is a simple dumb bell shape as shown in Fig. 2. There is one bottleneck link which has varying bandwidth with 10 ms delay. On one side of bottleneck link there are 50 web clients and 50 voice sources/sinks. On the other side there are 5 web servers and 50 voice sources/sinks. There are two best effort sources and sinks to produce congestion on the bottleneck link. There is minimum bandwidth reserved for the BE sources but these sources always send at the rate higher than the rate allocated to them.

Following three types of traffic are used in the network:

1. **Voice traffic**. The voice traffic is modeled as VoIP and there is no compression and silence suppression [1]. There are 50 voice source/sink pairs at each side of bottleneck link. The VoIP sources are actually UDP ON/OFF sources. The inter call gap is 15 minutes and the mean rate of traffic is 86.4 kbit/s. The 80% of the calls are short calls and rest are long calls. The on time for short calls is 3 minutes and that for long calls is 8 minutes. The VoIP traffic is assigned expedited forwarding (EF) PHB. EF PHB provides low latency, low loss, low jitter, assured bandwidth service through DiffServ domains [4].
2. **Data traffic**. The data traffic is web traffic generated by the request and reply interaction of HTTP/1.1 between web servers and clients. There are 50 clients requesting to 5 web servers [1]. The number of objects requested are random. This traffic is assigned to assured forwarding PHB. AF PHB provides forwarding assurance to the packets belonging to this PHB [3].
3. **Best effort**. The best effort (BE) traffic is a simple UDP source generating at the rate higher than the rate it is allowed. The traffic assigned to BE PHB has no assurance from DiffServ domain.

6.2. Simulation results

The end-to-end delay and packet loss for different classes are measured. These values vary with the change of the capacity of the bottleneck link. The results are shown in two different ways. There are tables in Section 6.2.1 that show the packet loss for traffic belonging to all PHBs. The graphs in Section 6.2.2 show the average end-to-end delay.

6.2.1. Packet statistics

Tables 1 and 2 show the packet loss statistics of the traffic of different classes. In Tables CP is the DiffServ code point of the packet. TotPkts and TxPkts are the counters of packets received and packets transmitted respectively. The ldrops are the packets that are dropped due to link overflow. Edrops mean the packets dropped due to random early detection (RED) early dropping mechanism. The code point 10, 20 and 30 are for the traffic belonging to EF, AF and BE classes respectively. The code point 21 is assigned to out-of-profile packets of AF class.

Table 1
Packets statistics at router 2: bandwidth 2.0 Mbit/s

CP	TotPkts	TxPkts	ldrops	edrops
All	832 524	341 706	490 818	0
10	80 543	73 125	7 418	0
20	173 118	172 115	1 003	0
21	78 863	0	78 863	0
30	500 000	96 466	403 534	0

Table 2
Packets statistics at router 2: bandwidth 3.0 Mbit/s

CP	TotPkts	TxPkts	ldrops	edrops
All	842 908	462 496	380 412	0
10	82 617	82 615	2	0
20	171 882	171 672	210	0
21	88 409	0	88 409	0
30	500 000	208 209	291 791	0

Tables 1 and 2 show the packet statistics of the traffic at router 2 whereas the bottleneck link capacity is 2.0 Mbit/s and 3.0 Mbit/s respectively. By comparing the tables it is obvious that the number of dropped packets reduces considerably with the increase of the link capacity. If same tables are to be used by BB to define PDB then BB may interpret those in the following manner:

1. For the specified load conditions the link with bandwidth of 2.0 Mbit/s has packet drop of almost 10% for EF traffic.
2. For the same load conditions the packet drop for EF traffic for the link with bandwidth of 3.0 Mbit/s is less than 1%.
3. If the SLA with user requires packet loss less than 1% then link capacity should be 3.0 Mbit/s.
4. BB can indicate during SLA negotiation that the packet loss for EF traffic is less than 1%.

6.2.2. End-to-end delay

The graphs presented in this section show end-to-end delay for VoIP and best effort traffic during the simulation. In Figs. 3 and 4 along *x*-axis is the time in seconds and *y*-axis has the average end-to-end delay in seconds. From the graphs, it can be observed that in the beginning of the simulation, the delay is lower but as more and more sources start sending traffic the average delay increases. The end-to-end delay mentioned here is the average of all the sources belonging to that particular PHB.

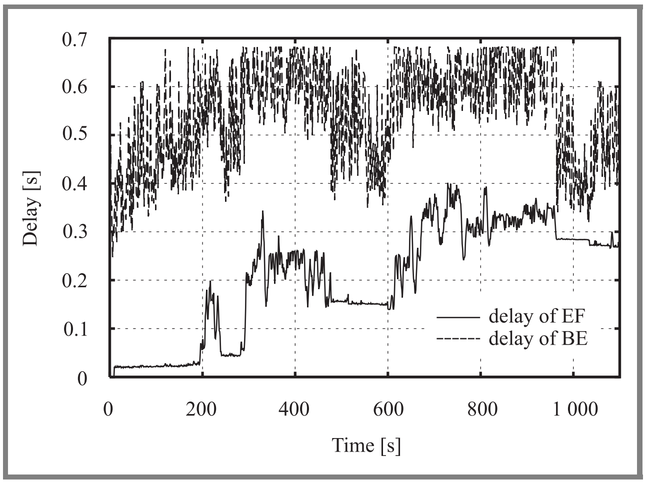


Fig. 3. End-to-end delay of EF and BE PHB when the capacity of the bottleneck link is 2.0 Mbit/s.

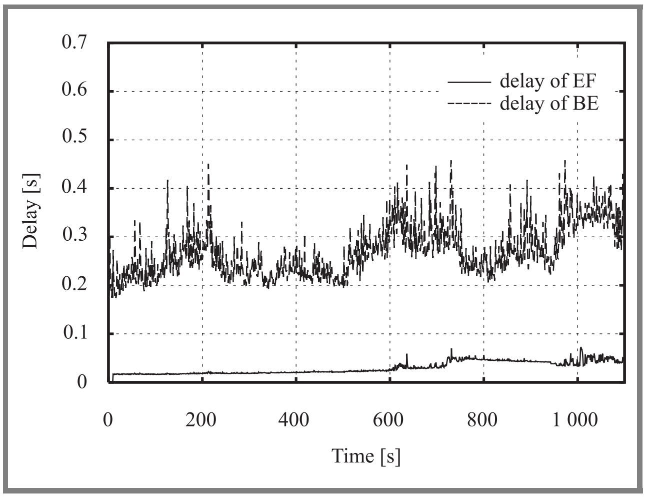


Fig. 4. End-to-end delay of EF and BE PHB when the capacity of the bottleneck link is 3.0 Mbit/s.

Figures 3 and 4 show the average end-to-end delay of EF and BE traffic when the capacity of the bottleneck link is 2.0 Mbit/s and 3.0 Mbit/s respectively. By comparing Figs. 3 and 4 it can be seen that the average end-to-end delay reduces considerably from 280 ms to less than 50 ms with increase of bandwidth. BB may interpret these results in order to calculate PDB in the following manner:

1. For the specified load conditions, the link with bandwidth of 2.0 Mbit/s has average end-to-end delay of almost 280 ms for EF traffic.
2. For the same load conditions the average end-to-end delay for EF traffic for the link with bandwidth of 3.0 Mbit/s is less than 50 ms.
3. If user SLA requires average delay less than 50 ms then link capacity should be 3.0 Mbit/s.
4. BB can indicate during SLA negotiation that the edge-to-edge average delay for EF traffic is less than 50 ms in its domain.

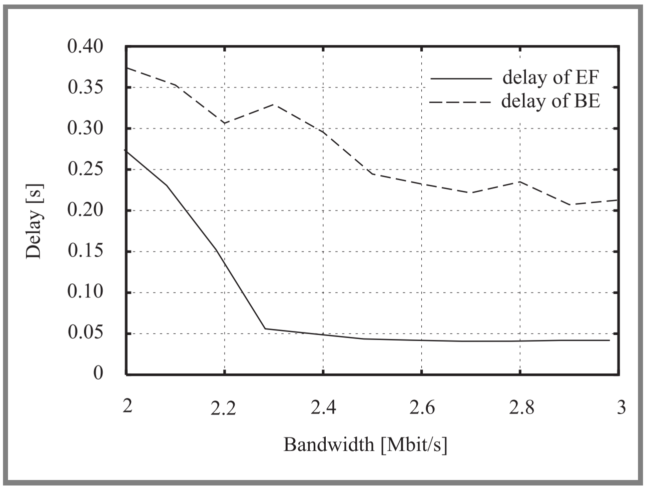


Fig. 5. End-to-end delay of EF and BE PHB with the variation of capacity of bottleneck link.

Figure 5 shows the variation of end-to-end delay with the variation of capacity of bottleneck link. Along *y*-axis is the average delay in seconds and along *x*-axis is the link capacity in Mbit/s. This type of graph can give an idea as how much delay can be accepted when the traffic passes through a specified link at a particular load.

6.3. Discussion

It is evident from the graphs and the tables presented in the previous subsections that by using a simple approach like this, BB can find QoS attributes for PDB of different PHBs. BB may choose to specify the range of these QoS parameters that can be supported by the domain.

The packet statistics tables show the number of packets lost for every type of PHB. These values can be used to perform off-line analysis by BB to find out the minimum bandwidth required to support some specific packet loss value for particular PHB. BB may get these packet loss statistics at different time of the day or month. These statistics can perform important role in performing future capacity planning.

The end-to-end delay is a very important QoS parameter for optimal performance of some applications. Calculating it with a simple mechanism used in the simulations can greatly reduce the overhead. We have only calculated the average delay however calculating maximum or minimum delay with the same mechanism is a trivial task. BB may use these values to specify the range of delay that particular PHB can suffer. BB can perform an efficient analysis of these values for future capacity planning as well as efficient QoS guarantees.

7. The impact

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 become the bottleneck to allocate the network resources effectively even in the scenarios when the network is underutilized.

The extension of BB to optimize the resource allocation by using PDB values can increase the BB's complexity exponentially. The approach discussed in this paper greatly depends upon the ability of BB to perform efficient offline analysis of the PDB information. By adding the ability to BB to analyze PDB information offline, the already overburden BB does not have to perform any additional processing for optimal SLA negotiation and resource allocation.

The scalability issues of BB are dealt by introducing the concept of multiple BB architecture [18] and multi layer BB architecture [19]. In the multiple BB architecture there is one central BB and number of edge BBs in the domain. The total domain resources are divided among edge BBs and it is the responsibility of the edge BB to efficiently manage the resources allocated to it. The central BB is responsible for overall resource management and coordination among edge BBs. The multi layer BB architecture distributes the functionality of BB among multiple entities and some of those entities are structured hierarchically to further break down the scalability problem. The BB's extension to optimize the resource allocation by using PDB information can be added to multi layer and multiple BB architecture. However, in both the architectures the information analysis should be performed in a way that the analysis results are updated at all resource management entities at the same time.

8. Conclusion and future work

An idea of using BB to measure and calculate attributes of PDB for dynamic SLA negotiation is proposed in this paper. Simulation was performed to give idea about the mechanism that can be used to relate these attributes to parameters of the network.

Introducing this type of mechanism in BB can increase its complexity, however the magnitude of this complexity entirely depends upon the ISPs. During SLA negotiation these attributes of PDB for different PHBs can give ISP an edge over others in defining their services better and in the terms that are better understood by users. Moreover ISPs can use this mechanism in their domain's BB to provide extra motivation to the user to select their services.

The DiffServ working group has defined PDBs but how, when and where to calculate and advertise these are the topics for future research. We have presented a simulation study to elaborate our idea of adding the ability of calculating PDBs in BB. We are planning to do more simulation studies in this area using complex topologies and calculating more PDB attributes.

References

[1] U. Fiedler, P. Huang, and B. Plattner, "Towards provisioning Diff-Serv intranets", in *Lecture Notes in Computer Science*, 2001, vol. 2092, pp. 27–43.

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

[3] J. Heinanen *et al.*, "Assured forwarding PHB group", Internet request for comments RFC2597, IETF, June 1999.

[4] V. Jacobson *et al.*, "An expedited forwarding PHB", Internet request for comments RFC2598, IETF, June 1999.

[5] V. Jacobson *et al.*, "The virtual wire per-domain behavior", Internet draft, IETF, July 2000.

[6] G. Kim, P. Mouchtaris, S. Samtani, R. Talpade, and L. Wong, "QoS provisioning for VoIP in bandwidth broker architecture: a simulation approach", in *Commun. Netw. Distrib. Syst. Model. Simul. Conf. CNDS* '01, Phoenix, USA, Jan. 2001.

[7] K. Nichols and B. Carpenter, "Definition of differentiated services per domain behaviors and rules for their specification", Internet request for comments RFC3086, IETF, Apr. 2001.

[8] K. Nichols *et al.*, "A lower effort per-domain behavior for differentiated services", Internet draft, IETF, June 2002.

[9] N. Seddigh *et al.*, "An assured rate per-domain behavior for differentiated services", Internet draft, IETF, Feb. 2001.

[10] Network simulator NS-2, Jan. 2003, http://www.isi.edu/nsnam/ns/

[11] S. Sohail and S. Jha, "The survey of bandwidth broker", Tech. Rep. UNSW CSE TR 0206, School of Computer Science and Engineering, University of New South Wales, Sydney, Australia, May 2002.

[12] B. Teitelbaum and P. Chimento, "Qbone bandwidth broker architecture", Work in Progress, June 2002, http://qbone.ctit.utwente.nl/deliverables/1999/d2/bboutline2.htm

[13] B. Teitelbaum and R. Geib, "Internet2 QBone: a test bed for differentiated service", in *INET '99*, *Internet Glob. Sum.*, San Jose, USA, June 1999.

[14] "WFQ scheduler for NS-2", June 2002, http://www.tik.ee.ethz.ch/fiedler/provisioning.html

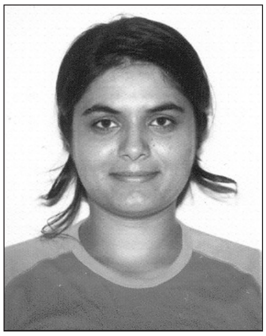
[15] T. Braun, M. Gunter, I. Khalil, R. Balmer, and F. Baumgartner, "Virtual private network and quality of service management implementation", Tech. Rep. IAM-99-003, CATI, July 1999.

[16] T. Braun and I. Khalil, "Edge provisioning and fairness in VPN-DiffServ networks", in *9th Int. Conf. Comput. Commun. Netw. ICCCN 2000*, pp. 424–433.

[17] T. Braun and I. Khalil, "A range based SLA and edge driven virtual core provisioning in DiffServ-VPNs", in *26th Ann. IEEE Conf. Loc. Comp. Netw. LCN'2001*, Tampa, USA, 2001.

[18] Z. Zhang, Z. Duan, and Y. Hou, "On scalable design of bandwidth brokers", in *IEICE Trans. Commun.*, E84-B (8), 2001.

[19] G. Politis, P. Sampatakos, and I. Venieris, "Design of a multi-layer bandwidth broker architecture", in *Interworking*, 2000, pp. 316–325.



Shaleeza Sohail is a Ph.D. student in the School of Computer Science and Engineering at the University of New South Wales (UNSW), Sydney, Australia. She has her masters degree from UNSW in 2000. She is a member of Network Research Laboratory of UNSW and currently being supervised by dr Sanjay Jha. Currently she is working on the bandwidth broker's extensions to provide network level QoS to next generation applications.
e-mail: sohails@cse.unsw.edu.au
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia

Sanjay Jha is an Associate Professor (Networks) at the School of Computer Science and Engineering (CSE) at the University of New South Wales. He has a Ph.D. degree from the University of Technology, Sydney, Australia. He is the founder of the Network Research Laboratory at CSE, UNSW. His research activities cover a wide range of topics in networking including quality of service, mobile/wireless Internet, and active/programmable network. He is the principal author of the book *Engineering Internet QoS* (Artech House, 2002). He has been working as an industry consultant for major organizations such as Canon Research Lab (CISRA), Lucent and Fujitsu. In his previous job, he was a lecturer at the School of Computing Sciences, University of Technology (UTS), Sydney. He also worked as systems engineer for the National Informatics Centre, New Delhi. He was a visiting scholar at the Distributed Computing and Communications Laboratory, Computer Science Department, Columbia University, New York, and Transmission Systems Development Department of Fujitsu Australia Ltd., Sydney.
e-mail: sjha@cse.unsw.edu.au
e-mail: jhask@ieee.org
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia