Achieving End-to-end QoS in the Next Generation Internet: Integrated Services over Differentiated Service Networks [1]

Haowei Bai and Mohammed Atiquzzaman [2]

Department of Electrical and Computer Engineering

The University of Dayton, Dayton, Ohio 45469-0226

E-mail: baihaowe@flyernet.udayton.edu, atiq@ieee.org

William Ivancic

NASA Glenn Research Center

21000 Brookpark Rd. MS 54-8,

Cleveland, OH 44135

Abstract

Currently there are two approaches to provide Quality of Service (QoS) in the next generation Internet: An early one is the Integrated Services (IntServ) with the goal of allowing end-to-end QoS to be provided to applications; the other one is the Differentiated Services (DiffServ) architecture providing QoS in the backbone. In this context, a DiffServ network may be viewed as a network element in the total end-to-end path. The *objective* of this paper is to investigate the possibility of providing end-to-end QoS when IntServ runs over DiffServ backbone in the next generation Internet. Our results show that the QoS requirements of IntServ applications can be successfully achieved when IntServ traffic is mapped to the DiffServ domain in next generation Internet.

[1]The work reported in this project was supported by NASA grant no. NAG3-2318

[2]The second author is currently with the School of Computer Science, University of Oklahoma, Norman, OK 73072, Tel: (405) 325 8077, email: atiq@ou.edu

1 Introduction

Quality of Service (QoS) has become the objective of the next generation Internet. QoS is generally implemented by different classes of service contracts for different users. A service class may provide low-delay and low-jitter services for customers who are willing to pay a premium price to run high-quality applications, such as, real-time multimedia. Another service class may provide predictable services for customers who are willing to pay for reliability. Finally, the *best-effort* service provided by current Internet will remain for those customers who need only connectivity.

The Internet Engineering Task Force (IETF) has proposed a few models to meet the demand for QoS. Notable among them are the Integrated Services (IntServ) model [1] and Differentiated Services (DiffServ) [2] model. The IntServ model is characterized by resource reservation. Before data is transmitted, applications must set up paths and reserve resources along the path. The basic target of the evolution of IntServ is to support various applications with different levels of QoS within the TCP/IP (Transport Control Protocol/Internet Protocol) architecture. But IntServ implementation requires RSVP (Resources Reservation Protocol) signaling and resource allocations at every network element along the path. This imposes a bound on its incorporation for the entire Internet backbone.

The DiffServ model is currently being standardized to overcome the above scalability issue, and to accommodate the various service guarantees required for time critical applications. The DiffServ model utilizes six bits in the TOS (Type of Service) field of the IP header to mark a packet for being eligible for a particular forwarding behavior. The model does not require significant changes to the existing infrastructure, and does not need many additional protocols. Therefore, with the implementation of IntServ for small WAN networks and DiffServ for the Internet backbone, the present TCP/IP traffic can meet the present day demands of real time and other quality required traffic. Combining IntServ and DiffServ has been proposed by IETF in [3] [4] as one of the possible

solutions to overcome the scalability problem.

To combine the advantages of DiffServ (good scalability in the backbone) and IntServ (per flow QoS guarantee), a mapping from IntServ traffic flows to DiffServ classes has to be performed. Some preliminary work has been carried out in this area. Authors in [5] present a concept for the integration of both IntServ and DiffServ, and describe a prototype implementation using commercial routers. However, they don't present any numerical results. Authors in [6] present results to determine performance differences between IntServ and DiffServ, as well as some characteristics about their combined use.

The *objective* of this paper is to investigate the end to end QoS that can be achieved when IntServ runs over the DiffServ network in the next generation Internet. Our *approach* is to add a mapping function to the edge DiffServ router so that the traffic flows coming from IntServ domain can be appropriately mapped into the corresponding *Behavior Aggregates* of DiffServ, and then marked with the appropriate DSCP (Differentiated Service Code Point) for routing in the DiffServ domain. We show that, without making any significant changes to the IntServ or DiffServ infrastructure and without any additional protocols or signaling, it is possible to provide QoS to IntServ applications when IntServ runs over a DiffServ network.

The *significance* of this work is that end-to-end QoS over heterogeneous networks could be possible if the DiffServ backbone is used to connect IntServ subnetworks in the next generation Internet. The main *contributions* of this paper can be summarized as follows:

- Propose a mapping function to run IntServ over the DiffServ backbone.
- Show that QoS can be achieved by end IntServ applications when running over DiffServ backbone in the next generation Internet.

The rest of this paper is organized as follows. In Sections 2 and 3, we briefly present the

main features of IntServ and DiffServ, respectively. In Section 4, we describe our approach for the mapping from IntServ to DiffServ and the simulation configuration to test the effectiveness of our approach. In Section 5, we analyze our simulation results to show that QoS can be provided to end applications in the IntServ domain. Concluding remarks are finally given in Section 6.

2 Integrated Services

The Integrated Services (IntServ) model [1] characterized by resource reservation defines a set of extensions to the traditional *best effort* model with the goal of providing end-to-end QoS to applications. This architecture needs some explicit signaling mechanism to convey information to routers so that they can provide requested services to flows that require them. RSVP is one of the most widely known example of such a signaling mechanism. We will describe this mechanism in details in Section 2.2. In addition to the *best effort* service, the integrated services model provides two service levels as follows.

- *Guaranteed service* [7] for applications requiring firm bounds on end-to-end datagram queueing delays.
- *Controlled-load service* [8] for applications requiring services closely equivalent to that provided to uncontrolled *best effort* traffic under unloaded (lightly loaded) network conditions.

We will discuss them in Sections 2.3 and 2.4, respectively.

2.1 Components of Integrated Services

The basic framework of integrated services [4] is implemented by four components: the *signaling protocol* (e.g., RSVP), the *admission control routine*, the *classifier* and the *packet scheduler*. In this model, applications must set up paths and reserve resources before transmitting their data. Network

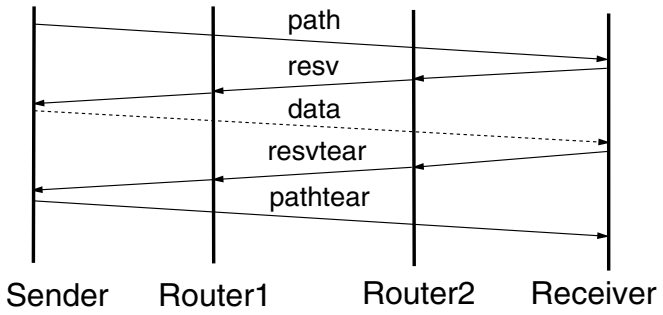

Figure 1: RSVP signaling for resource reservation.

elements will apply admission control to those requests. In addition, traffic control mechanisms on the network element are configured to ensure that each admitted flow receives the service requested in strict isolation from other traffic. When a router receives a packet, the classifier will perform a MF (multifield) classification and put the packet in a specific queue. The packet scheduler will then schedule the packet according to its QoS requirements.

2.2 RSVP Signaling

RSVP is a signaling protocol to reserve network resources for applications. Figure 1 illustrates the setup and teardown procedures of PSVP protocol. The sender sends a **PATH** message to the receiver specifying the characteristic of the required traffic. Every intermediate router along the path forwards the **PATH** message to the next hop determined by the routing protocol. If the receiver agrees the advertised flow, it sends a **RESV** message, which is forwarded hop by hop via RSVP capable routers towards the sender of the **PATH** message. Every intermediate router along the path may reject or accept the request. If the request is accepted, resources are allocated, and **RESV** message is forwarded. If the request is rejected, the router will send an **RESV-ERR** message back to the sender of the **RESV** message.

If the sender gets the **RESV** message, it means resources are reserved and data can be transmitted. To terminate a reservation, a **RESV-TEAR** message is transmitted to remove the resource

allocation and a **PATH-TEAR** message is sent to delete the path states in every router along the path.

2.3 Guaranteed Service

Guaranteed service guarantees that datagrams will arrive within the guaranteed delivery time and will not be discarded due to queue overflows, provided the flow's traffic stays within its specified traffic parameters [7]. The service provides assured level of bandwidth or link capacity for the data flow. It imposes a strict upper bound on the end-to-end queueing delay as data flows through the network. The packets encounter no queueing delay as long as they conform to the flow specifications. It means packets cannot be dropped due to buffer overflow and they are always guaranteed the required buffer space. The delay bound is usually large enough even to accommodate cases of long queueing delays.

2.4 Controlled-load Service

The controlled-load service does not accept or make use of specific target values for control parameters such as delay or loss. Instead, acceptance of a request for controlled-load service is defined to imply a commitment by the network elements to provide the requester with a service closely equivalent to that provided to uncontrolled (best effort) traffic under lightly loaded conditions [8]. The service aims at providing the same QoS under heavy loads as under unloaded conditions. Though there is no specified strict bound on delay, it ensures that very high percentage of packets do not experience delays highly greater than the minimum transit delay due to propagation and router processing.

3 Differentiated Services

The IntServ/RSVP architecture described in Section 2 can be used to provide QoS to applications. All the routers are required to be capable of RSVP, admission control, MF classification and packet scheduling, which needs to maintain all the information for each flow at each router. The above issues raise scalability concerns in large networks [4]. Because of the difficulty in implementing and deploying integrated services and RSVP, differentiated services is currently being developed by the IETF [2].

Differentiated services (DiffServ) is intended to enable the deployment of scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. The premise of DiffServ networks is that routers in the core network handle packets from different traffic streams by forwarding them using different per-hop behaviors (PHBs). The PHB to be applied is indicated by a DiffServ Codepoint (DSCP) in the IP header of the packet [9]. The advantage of such a mechanism is that several different traffic streams can be aggregated to one of a small number of behavior aggregates (BA) which are each forwarded using the same PHB at the router, thereby simplifying the processing and associated storage [10]. There is no signaling or processing since QoS (Quality of Service) is invoked on a packet-by-packet basis [10].

The DiffServ architecture is composed of a number of functional elements, including a small set of per-hop forwarding behaviors, packet classification functions, and traffic conditioning functions which includes metering, marking, shaping and policing. The functional block diagram of a typical DiffServ router is shown in Figure 2 [10]. This architecture provides *Expedited Forwarding* (EF) service and *Assured Forwarding* (AF) service in addition to *best-effort* (BE) service as described below.

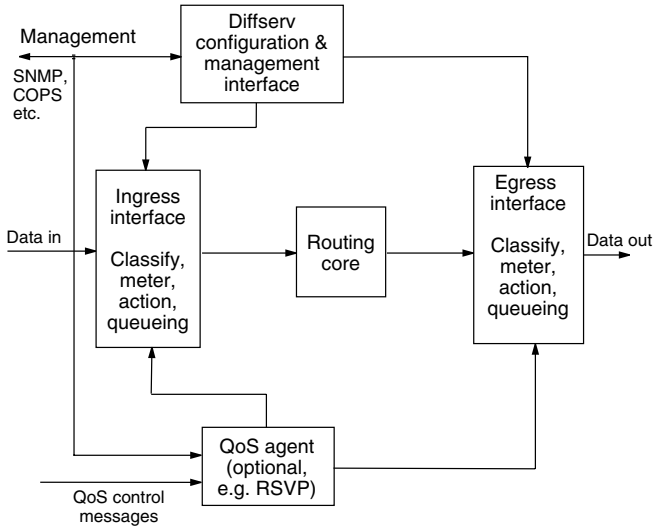

Figure 2: Major functional block diagram of a router.

3.1 Expedited Forwarding (EF)

This service is also been described as *Premium Service*. The EF service provides a low loss, low latency, low jitter, assured bandwidth, end-to-end service for customers [11]. Loss, latency and jitter are due to the queuing experienced by traffic while transiting the network. Therefore, providing low loss, latency and jitter for some traffic aggregate means there are no queues (or very small queues) for the traffic aggregate. At every transit node, the aggregate of the EF traffic's maximum arrival rate must be less than its configured minimum departure rate so that there is almost no queuing delay for these premium packets. Packets exceeding the peak rate are shaped by the traffic conditioners to bring the traffic into conformance.

3.2 Assured Forwarding

This service provides a reliable services for customers, even in times of network congestion. Classification and policing are first done at the edge routers of the DiffServ network. The assured service traffic is considered *in*-profile if the traffic does not exceed the bit rate allocated for the service; otherwise, the excess packets are considered *out*-of-profile. The *in*-profile packets should be forwarded

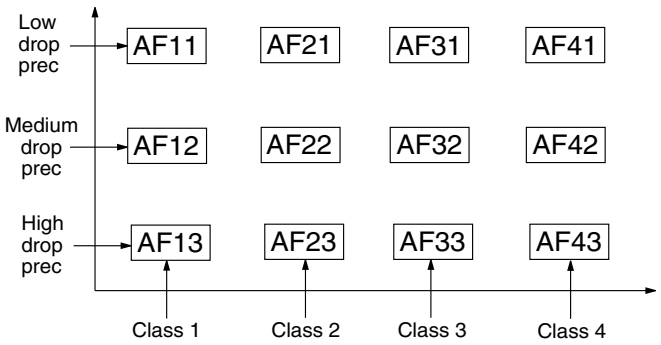

Figure 3: AF classes with drop precedence levels.

with high probability. However, the *out*-of-profile packets are not delivered with as high probability as the traffic that is within the profile. Since the network does not reorder packets that belong to the same microflow, all packets, irrespective of whether they are *in*-profile or *out*-of-profile, are put into an *assured queue* to avoid out-of-order delivery.

Assured Forwarding provides the delivery of packets in four independently forwarded AF classes. Each class is allocated with a configurable minimum amount of buffer space and bandwidth. Each class is in turn divided into different levels of drop precedence. In the case of network congestion, the drop precedence determines the relative importance of the packets within the AF class. Figure 3 [12] shows four different AF classes with three levels of drop precedence.

3.3 Best Effort

This is the default service available in DiffServ, and is also deployed by the current Internet. It does not guarantee any bandwidth to the customers, but can only get the bandwidth available. Packets are queued when buffers are available and dropped when resources are over committed.

4 Integrated Services over Differentiated Services Networks

In this section, we describe in details the mapping strategy adopted in this paper to connect the IntServ and DiffServ domains. Simulation configuration that has been used to test the mapping strategy is described in 4.3 .

4.1 Mapping Considerations for IntServ over DiffServ

In IntServ, resource reservations are made by requesting a service type specified by a set of quantitative parameters known as *Tspec* (Traffic Specification). Each set of parameters determines an appropriate priority level. When requested services with these priority levels are mapped to DiffServ domain, some basic requirements should be satisfied.

- PHBs in DiffServ domain must be appropriately selected for each requested service in IntServ domain.
- The required policing, shaping and marking must be done at the edge router of the DiffServ domain.
- Taking into account the resource availability in DiffServ domain, admission control must be implemented for requested traffic in IntServ domain.

4.2 Mapping Function

The mapping function is used to assign an appropriate DSCP to a flow specified by *Tspec* parameters in IntServ domain, such that the same QoS could be achieved for IntServ when running over DiffServ domain. Each packet in the flow from the IntServ domain has a *flow ID* indicated by the value of *flow-id* field in the IP (Internet Protocol) header. The *flow ID* attributed with the *Tspec* parameters is used to determine which flow the packet belongs to. The main constraint is that the

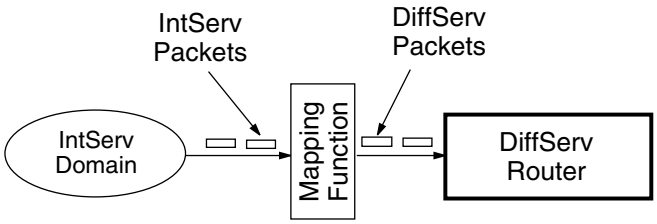

Figure 4: Mapping function for integrated service over differentiated service.

PHB treatment of packets along the path in the DiffServ domain must approximate the QoS offered by IntServ itself. In this paper, we satisfy the above requirement by appropriately mapping the flows coming from IntServ domain into the corresponding *Behavior Aggregates*, and then marking the packets with the appropriate DSCP for routing in the DiffServ domain.

To achieve the above goal, we introduce a mapping function at the boundary router in DiffServ domain as shown in Figure 4. Packets specified by *Tspec* parameters in IntServ domain are first mapped to the corresponding PHBs in the DiffServ domain by appropriately assigning a DSCP according to the mapping function. The packets are then routed in the DiffServ domain where they receive treatment based on their DSCP code. The packets are grouped to BAs in the DiffServ domain. Table 1 shows an example mapping function which has been used in our simulation. As an instance, a flow in IntServ domain specified by *r=0.7Mb, b=5000bytes* and *Flow ID=0* is mapped to *EF* PHB (with corresponding DSCP 101110) in DiffServ domain, where r means token bucket rate and b means token bucket depth.

Table 1: An example mapping function used in our simulation.

Tspec	*Flow ID*	*PHB*	*DSCP*
r=0.7 Mb, b=5000 bytes	0	EF	101110
r=0.7 Mb, b=5000 bytes	1	EF	101110
r=0.5 Mb, b=8000 bytes	2	AF11	001010
r=0.5 Mb, b=8000 bytes	3	AF11	001010
r=0.5 Mb, b=8000 bytes	4	AF11	001010

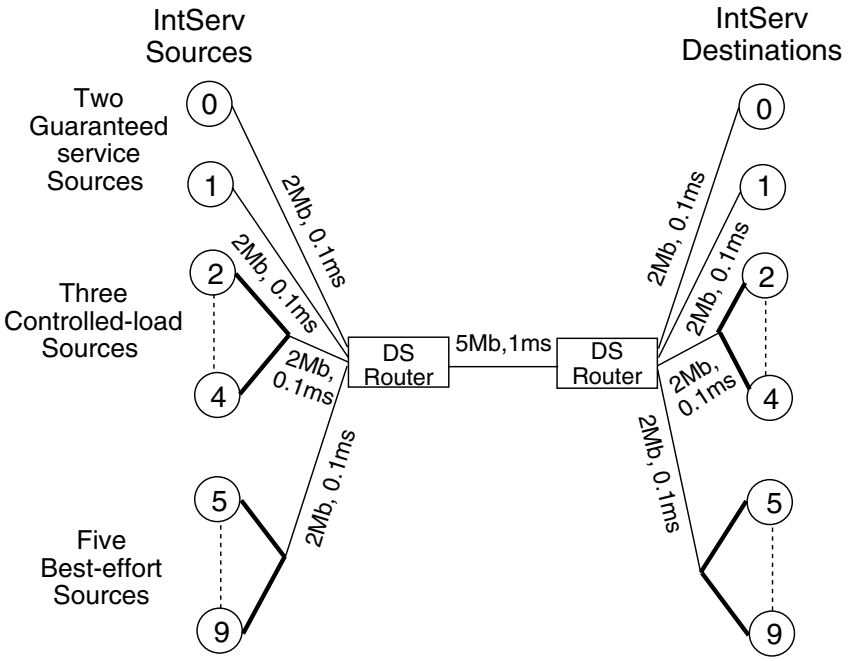

Figure 5: Network simulation configuration.

The sender initially specifies its requested service using *Tspec*. Note that it is possible for different senders to use the same *Tspec*. However, they are differentiated by the *flow ID*. In addition, it is also possible that different flows can be mapped to the same PHB in DiffServ domain.

4.3 Simulation Configuration

To test the effectiveness of our proposed mapping strategy between IntServ and DiffServ and to determine the QoS that can be provided to IntServ applications, we carried out simulation using the *ns* (Version 2.1b6) simulation tool from Berkeley [13]. The network configuration used in our simulation is shown in Figure 5.

Ten IntServ sources were used in our simulation, the number of sources generating *Guaranteed services*, *Controlled-load* services and *best-effort* services were two, three and five respectively. Ten IntServ sinks served as destinations for the IntServ sources. We set the *flow IDs* to be the same as the corresponding source number shown in Figure 5.

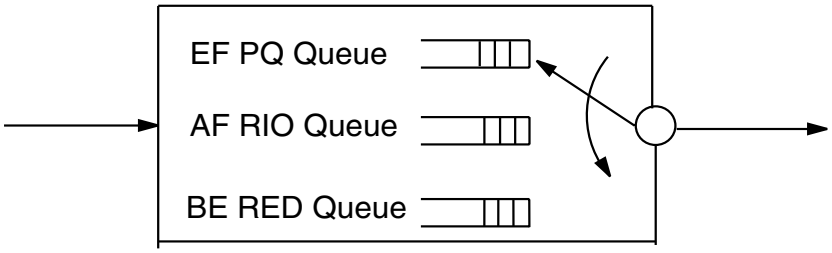

Figure 6: Queues inside the edge DiffServ router.

All the links in Figure 5 are labeled with a (*bandwidth, propagation delay*) pair. The mapping function shown in Table 1 has been integrated into the DiffServ edge router (See Figure 4). CBR (Constant Bit Rate) traffic was used for all IntServ sources in our simulation so that the relationship between the bandwidth utilization and bandwidth allocation can be more easily evaluated. *Note that ten admission control modules have been applied to each link between sources and DiffServ edge routers to guarantee the resource availability within DiffServ domain. To save space, they are not illustrated in Figure 5. Admission control algorithm was implemented by token bucket with parameters specified in Table 1.*

Inside the DiffServ edge router, EF queue was configured as a simple Priority Queue with Tail Drop; AF queue was configured as RIO queue and BE queue as a RED [14] queue, which are shown in Figure 6. The queue weights of EF, AF and BE queues were set to 0.4, 0.4 and 0.2 respectively. Since the bandwidth of the bottleneck link between two DiffServ routers is 5 Mb, the above scheduling weights implies bandwidth allocations of 2 Mb, 2 Mb and 1 Mb for the EF, AF and BE links respectively during periods of congestion at the edge router.

5 Simulation Results

In this section, results obtained from our simulation experiments are presented. The criteria used to evaluate our proposed strategy are first described followed by the explanations of our experimental

and numerical results.

5.1 Performance Criteria

To show the effectiveness of our mapping strategy in providing QoS to end IntServ applications, we have used *goodput*, *queue size* and *drop ratio* as the performance criteria. In addition, in order to prove the effectiveness of admission control mechanism, we also measured the *non-conformant ratio* (the ratio of non-conformant packets out of in-profile packets). In Section 5.2, we present the results of measurements of the above quantities from our simulation experiments.

5.2 QoS Obtained by Guaranteed Services

We use the following three simulation cases to determine the QoS obtained by IntServ applications. As results, Table 2 shows the goodput of each *Guaranteed service* source for three different cases described in Section 5.2. Table 3 shows the drop ratio measured at the scheduler for three cases of the *Guaranteed service* sources. Table 4 shows the non-conformant ratio for each *Guaranteed service* source. Figures 7, 8 and 9 show the queue size for each of the three case, from which the queuing delay and jitter can be evaluated.

Table 2: Goodput of each *Guaranteed service* source (Unit: Kb/S)

Tspec	*Flow ID*	*Case 1*	*Case 2*	*Case 3*
r=0.7 Mb, b=5000 bytes	*0*	699.8250	699.8039	459.8790
r=0.7 Mb, b=5000 bytes	*1*	699.8039	699.6359	1540.1400

Table 3: Drop ratio of *Guaranteed service* traffic.

Type of traffic	*Case 1*	*Case 2*	*Case 3*
Guaranteed service Traffic	0.000000	0.000000	0.258934

Table 4: The non-conformant ratio for each *Guaranteed service* source

Tspec	*Flow ID*	*Case 1*	*Case 2*	*Case 3*
r=0.7 Mb, b=5000 bytes	*0*	0.00026	0.00026	0.00026
r=0.7 Mb, b=5000 bytes	*1*	0.00026	0.22258	0.00040

5.2.1 Case 1: No congestion; no excessive traffic

The traffic generated by *Guaranteed service* sources (source *0* and source *1*) were set to 0.7 Mb and 0.7 Mb, respectively. In this case, the traffic rate is equal to the bucket rate (0.7 Mb, shown in Table 1), which means *there should not be any significant excessive IntServ traffic*. According to the network configuration described in Section 4.3, two *Guaranteed service* sources generate 1.4 Mb traffic which is less than the corresponding scheduled link bandwidth for *Guaranteed service* (EF in DiffServ domain) traffic (2Mb). Under this scenario, *there should not be any significant congestion* at the edge DiffServ router.

Case 1 is an ideal case. As seen in Table 2, the goodput is almost equal to the corresponding source rate. From Table 3, since there is no significant congestion, the drop ratio of each type of sources is zero. Table 4 shows the performance of admission control mechanism. Since there is no excessive traffic in this case, the non-conformant ratio is almost zero. Figure 7 shows the queuing performance of each queue. Because this is an ideal case, the size of each queue is very small. Though the three queues have almost the same average size, we observe that the BE queue of IntServ (mapping to BE queue in DiffServ domain, according to the mapping function) has the largest jitter.

5.2.2 Case 2: No congestion; Guaranteed service source *1* generates excessive traffic

The traffic generated by *Guaranteed service* sources (source *0* and source *1*) were set to 0.7 Mb and 0.9 Mb, respectively. In this case, the traffic rate of source *1* is greater than its corresponding bucket

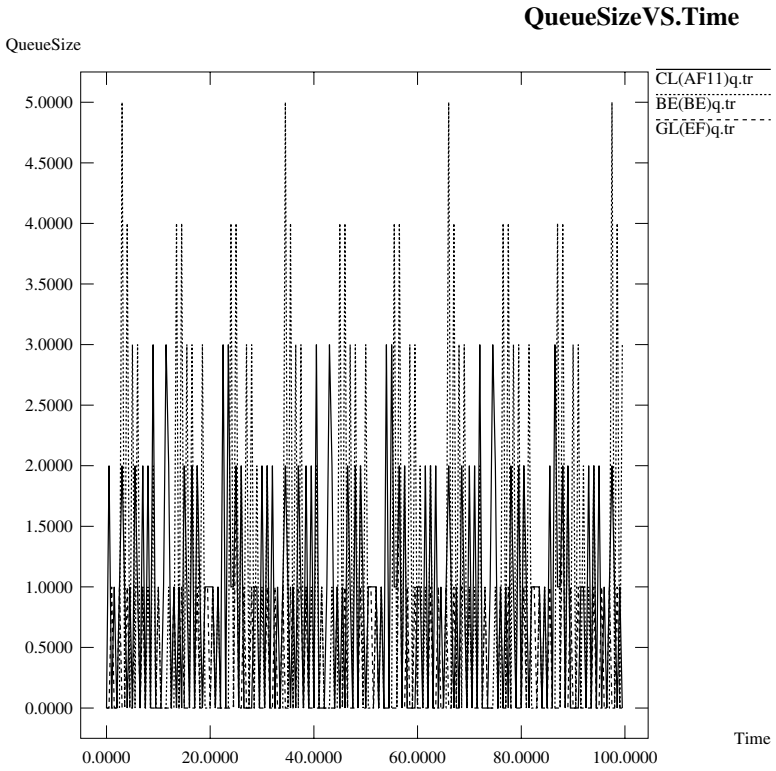

Figure 7: Queue size plots for *Case 1*.

rate (0.7 Mb, shown in Table 1), which means *source* 1 *generates excessive IntServ traffic*. According to the network configuration described in Section 4.3, two *Guaranteed service* sources generate 1.6 Mb traffic which is less than the corresponding scheduled link bandwidth for *Guaranteed service* (EF in DiffServ domain) traffic (2Mb). Under this scenario, *there should not be any significant congestion* at the edge DiffServ router.

In *case 2*, from Table 2, the goodput of source *0* is equal to its source rate. However, the goodput of source *1* is equal to the corresponding token rate, 0.7 Mb, rather than its source rate, 0.9 Mb. Table 3 shows that the drop ratio of *Guaranteed service* is *0*. The reason is that, in this case, there is no congestion for *Guaranteed service* traffic. Table 4 indicates how the admission control mechanism works. As seen in this table, the non-conformant packets ratio of source *1* is increased, compared to case 1. It is because source *1* generates excessive traffic in this case. From Figure 8, we find that the average queue size of the *best effort* queue is far greater than the other

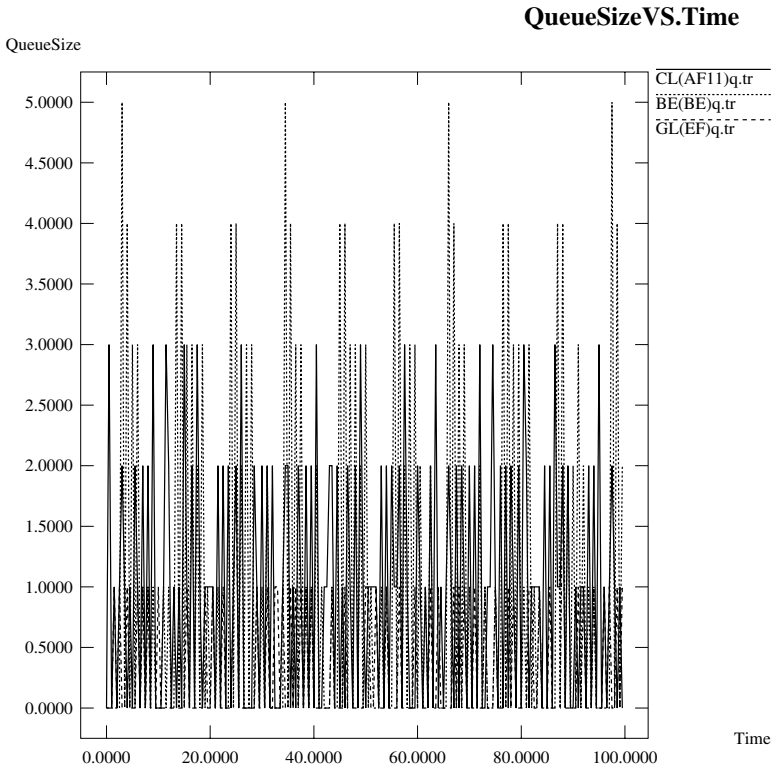

Figure 8: Queue size plots for *Case 2*.

two types of sources. In addition, the jitter of *best effort* traffic is also greater than the other two types of sources. The *Guaranteed service* traffic has the smallest average queue size and the smallest jitter. In addition, compared with Figure 7, the upper bound of *Guaranteed service* queue is guaranteed, though the source *1* generates more traffic than what it has reserved. This well satisfies requirements from [7].

5.2.3 Case 3: Guaranteed service gets into congestion; no excessive traffic

The traffic generated by *Guaranteed service* sources (source *0* and source *1*) were set to 0.7 Mb and 2 Mb, respectively. To simulate a congested environment, we set the token rate of source *1* to 2 Mb also. In this case, the traffic rate of source *1* is equal to its corresponding bucket rate (2 Mb), which means *there is no significant excessive IntServ traffic*. According to the network configuration described in Section 4.3, two *Guaranteed service* sources generate 2.7 Mb traffic which

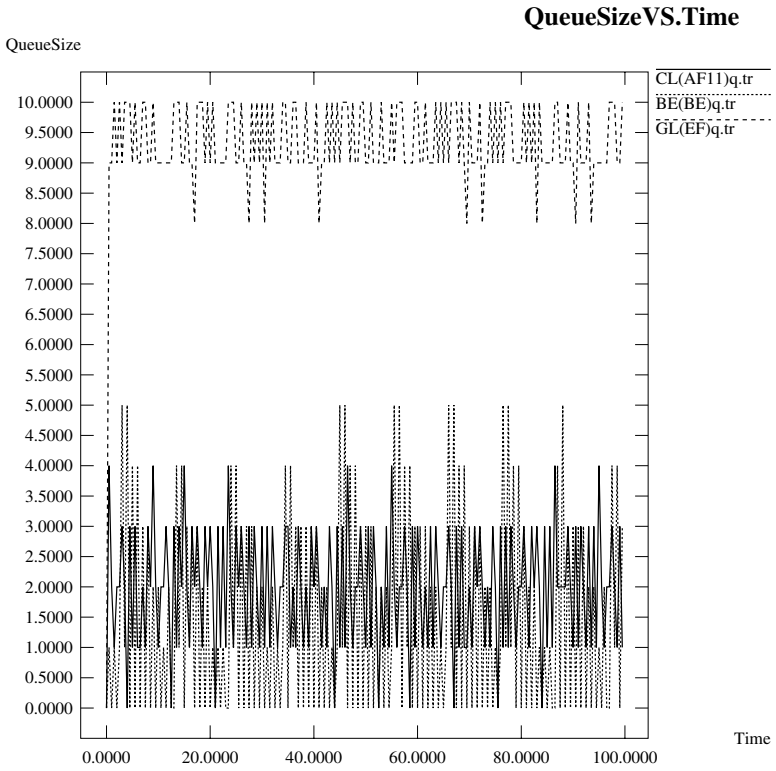

Figure 9: Queue size plots for *Case 3*.

is greater than the corresponding scheduled link bandwidth for *Guaranteed service* (EF in DiffServ domain) traffic (2Mb). Under this scenario, *Guaranteed service traffic gets into congestion* at the edge DiffServ router.

Case 3 is used to evaluate our mapping function under congested environments. As expected, we find the drop ratio (measured at scheduler) of *Guaranteed service* traffic is increased, and the total goodput of *Guaranteed service* is limited by the output link bandwidth assigned by the scheduler (2Mb), instead of 2.7 Mb. Since there is no excessive traffic, from Table 4, the no-conformant packets ratio of both of the *Guaranteed service* sources are closed to *0*. From Figure 9, since we increase the token rate of one of the *Guaranteed service* source (source *1*), the upper bound of *Guaranteed service* is increased, which is reasonable. In addition, the *Guaranteed service* queue still has the smallest jitter.

5.3 QoS Obtained by Controlled-load Services

Because of the similarity between the results of *Guaranteed service* and *Controlled-load service*, *all our descriptions in Section 5.2 are focused on* Guaranteed service. *We only give out results for Controlled-load service without detailed explanations.*

We use *case 2* described in Section 5.2.2 as an example. As described in Section 4.3, we used three *Controlled-load service* sources in our simulation: sources *2*, *3* and *4*. The token bucket parameters are shown in Table 1. We set the source rate of sources *2* and *4* to 0.5 Mb, 0.5 Mb, respectively, and set the rate of source *3* to 0.7 Mb (greater than its token rate, 0.5 Mb). Therefore, *source* 3 *generates excessive traffic*. The total *Controlled-load service* traffic is 1.7 Mb, which is less than the scheduled link bandwidth; therefore, *there should not be any significant congestion*.

Table 5 shows the goodput of each *Controlled-load* source. Table 6 shows the drop ratio of *Controlled-load service* measured at scheduler. Table 7 shows the non-conformant ratio. Figure 10 shows the queue size of this case. Note that though the non-conformant ratio of source *3* is much higher that the other two (shown in Table 7), the goodput of source*3* (shown in Table 5) is equal to its source rate (0.7 Mb). It is because the non-conformant packets are degraded and then forwarded, which is one of the forwarding schemes for non-conformant packets proposed by [8].

Table 5: Goodput of each *Controlled-load service* source (Unit: Kb/S)

Tspec	*Flow ID*	*Case 2*
r=0.5 Mb, b=8000 bytes	*2*	499.9889
r=0.5 Mb, b=8000 bytes	*3*	700.0140
r=0.5 Mb, b=8000 bytes	*4*	499.9889

Table 6: Drop ratio of *Controlled-load service* traffic.

Type of traffic	*Case 2*
Controlled-load Traffic	0.000000

Table 7: The non-conformant ratio for each *Controlled-load service* source

Tspec	*Flow ID*	*Case 2*
r=0.5 Mb, b=8000 bytes	*2*	0.00000
r=0.5 Mb, b=8000 bytes	*3*	0.28593
r=0.5 Mb, b=8000 bytes	*4*	0.00000

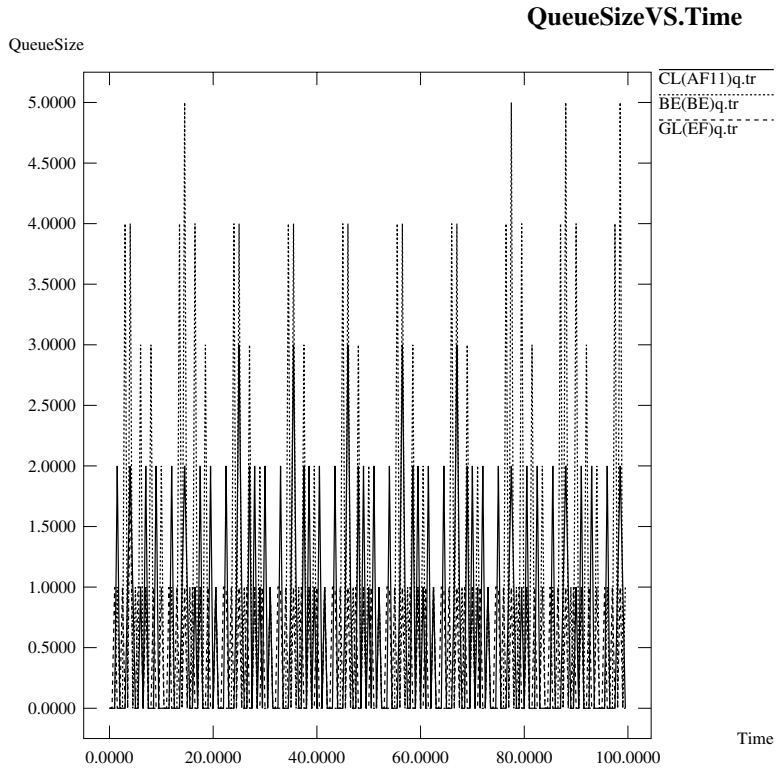

Figure 10: Queue size plots.

5.4 Observations

From the above results, we can arrive at the following *observations*:

- The upper bound of queueing delay of *Guaranteed service* is guaranteed. In addition, *Guaranteed service* always has the smallest jitter without being affected by other traffic flows, though [7] says it does not attempt to minimize the jitter. This well satisfies requirements from [7].

- The *Controlled-load service* has the smaller jitter and queue size than the *best effort* traffic. Furthermore, non-conformant packets are degraded and then forwarded, which is proved by

our simulation. This well satisfies requirements from [8].

We therefore, *conclude that the QoS requirements of IntServ can be successfully achieved when IntServ traffic is mapped to the DiffServ domain in next generation Internet.*

6 Conclusion

In this paper, we have proposed DiffServ as the backbone network to interconnect IntServ subnetworks. We have designed a mapping function to map traffic flows coming from IntServ with different priorities to the corresponding PHBs in the DiffServ domain.

The proposed scheme has been studied in detail using simulation. It has been found that the QoS requirements of IntServ can be achieved when IntServ subnetworks run over DiffServ. We have illustrated our scheme by mapping IntServ traffic of three different priorities to the three service classes of DiffServ. The ability of our scheme to provide QoS to end IntServ applications has been demonstrated by measuring the *drop ratio*, *goodput*, *non-conformant ratio* and *queue size*. We found that the upper bound of queueing delay of *Guaranteed service* is guaranteed. In addition, *Guaranteed service* always has the smallest jitter without being affected by other traffic flows, though [7] says it does not attempt to minimize the jitter. The *Controlled-load service* has the smaller jitter and queue size than the *best effort* traffic. Furthermore, non-conformant packets are degraded and then forwarded, which is proved by our simulation.

Acknowledgements

The authors thank NASA Glenn Research Center for support of this project. Jyotsna Chitri carried out some initial work on this project.

References

[1] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview." RFC 1633, June 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services." RFC 2475, December 1998.

[3] J. Wroclawski and A. Charny, "Integrated service mapping for differentiated services networks." draft-ietf-issll-ds-map-00.txt, March 2000.

[4] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine, "A framework for integrated services operation over diffserv networks." draft-ietf-issll-diffserv-rsvp-04.txt, March 2000.

[5] R. Balmer, F. Baumgarter, and T. Braun, "A concept for RSVP over diffserv," *Proc. Nineth ICCCN'2000*, Las Vagas, NV, October 2000.

[6] J. Harju and P. Kivimäki, "Co-operation and comparison of diffserv and intserv: Performance measurements," *Proc. 25th Conference on Local Computer Networks*, Tampa, FL, pp. 177–186, November 2000.

[7] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service." RFC 2212, September 1997.

[8] J. Wroclawski, "Specification of the controlled-load network element service." RFC 2211, September 1997.

[9] K. Nichols, S.Blake, F. Baker, and D.Black, "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers." RFC 2474, December 1998.

[10] Y. Bernet, S. Blake, D. Grossman, and A. Smith, "An informal management model for diffserv routers." draft-ietf-diffserv-model-04.txt, July 2000.

[11] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB." RFC 2598, June 1999.

[12] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group." RFC 2597, June 1999.

[13] VINT Project U.C. Berkeley/LBNL, "ns v2.1b6: Network simulator." http://www-mash.cs.berkeley.edu/ns/, January 2000.

[14] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.