대역폭 자원 할당을 위한 주문형 QoS 관리 시스템 구현

정회원 이 동 욱*, 준회원 이 동 훈*, 정회원 김 종 원*, 정 상 길**, 변 옥 환**

Implementation of On-demand QoS management System for Bandwidth Resource Allocation

Dongwook Lee* ***Regular Member***, DongHoon Yi* ***Associate Member***,
JongWon Kim¶*, Sanggil Jung** and Okhwan Byun** ***Regular Members***

요 약

인터넷상의 멀티미디어 응용들의 다양한 전송 특성을 지원하기 위해서는 모든 트래픽이 동등하게 처리되는 현재의 최선형 서비스 (best effort)를 극복하고 종단간 QoS를 보장해야 한다. 본 논문에서는 확장성 있는 IETF의 차등 서비스 (differentiated service: DiffServ)에 기반한 대역폭 중계자 (bandwidth broker) 형태의 QoS 할당 및 자원 관리 시스템을 구현한다. 구현된 시스템은 실시간 자원 할당, Edge/Core 라우터의 사용가능 자원 분석, 자원 예약, 시간 지연을 가지는 호수락기능, 사용자와 서버사이의 QoS협상 및 모니터링을 통한 자원 사용량 감시를 수행한다. 또한 실제 네트워크 상에서의 테스트베드를 통해 응용 트래픽의 요구에 따른 대역폭 할당을 수행함에 의해서 구현 시스템의 성능을 검증한다.

Key Words : Quality of Service (QoS), differentiated service(DiffServ), bandwidth broker (BB), virtual private network (VPN), service level agreement (SLA), and resource provisioning.

ABSTRACT

To support diverse transmission requirements of multimedia applications, Quality of Service (QoS) should be provided in the Internet, where only the best-effort service is available. In this paper, we describe our recent effort on the implementation and verification of an extendable and flexible QoS allocation and resource management system based on the bandwidth broker model for realizing the IETF differentiated service (DiffServ). Focusing on the bandwidth issue over single administrative domain, the implemented system provides real-time resource reservation and allocation, delayed call admission control, simple QoS negotiation between server and users, and simple resource monitoring. The implemented system is verified by evaluating the performance of a resource-intensive application over the real-world testbed network.

I. Introduction

Recently, Internet Applications (e.g., simple e-mail and web browsing) have been evolved into multimedia services such as VoIP (Voice over IP) and streaming services that require an enhanced QoS (quality of service) support. Real-time multimedia have different traffic characteristics and require various types of service quality. However,

*광주과학기술원 정보통신공학과 네트워크미디어연구실({dulee,dhyi,jongwon}@netmedia.kjist.ac.kr), **한국과학기술정보연구원
¶: 서신왕래저자
논문번호 : 030475-1027, 접수일자 : 2003년 월 일
※본 연구는 한국정보과학기술연구원(KISTI)의 지원으로 수행되었습니다.

current Internet provides only the best-effort delivery, where every type of traffic is treated the same. Thus, providing QoS support over the Internet has been a research challenge for many years. Firstly, Integrated Service (IntServ) has been proposed to support the end-to-end QoS for multimedia applications by the Internet Engineering Task Force (IETF). The IntServ can reserve a resource in routers along the routing path for a traffic flow by the help of Resource Reservation Protocol (RSVP). However, the IntServ suffers from the scalability problem; the state information of flows are so large that they cannot be processed in the core routers. On the other hand, the recent proposal from the IETF, Differentiated Service (DiffServ), proposes a scalable service discrimination model without requiring per-flow state management at the routers in the network core [1]. The DiffServ does not guarantee the QoS of each flow, and users flows are aggregated and processed based on the service classes. Processing complex traffic engineering at the edge routers while performing simple packet forwarding at the core routers makes the DiffServ architecture scalable and extensible in the large-scaled Internet.

The DiffServ adopts a *Bandwidth Broker* (BB) architecture as a resource allocation system. The BB is an agent that provides a centralized mechanism to control the resources within a single DiffServ domain. All agreements between clients and service providers which pertain to the type of service required are known as *Service Level Agreements* (SLA). The BB manages domain resources using service polices based on the SLA. At present many BB implementations are reported with slight variations in functionality. The implementations of real-time resource allocation systems over the bandwidth broker architecture are discussed in [3,4], and the hierarchical bandwidth broker systems, enhancing call admission capability, are proposed in [5-7]. Several experiments and results of BB systems over the DiffServ testbed such as the QBone of the Internet2 are reported in [3,4,8,9]. However, the implementations described above do not provide advance resource reservation and these run on a specific testbed where routing is configured as static. In our previous work, an on-demand QoS allocation system was implemented [2]. The purpose of the system was to provide the necessary infrastructure so that guaranteed QoS can be served on the supercomputer user network over the Korea Research Environment Open Network (KREONET). As a part of this project, a BB was developed and tested on the KREONET.

In this paper, we explain the implementation of an on-demand QoS allocation and resource management system based on the bandwidth broker architecture enhancing the previous prototype [2]. This QoS allocation system has requirements in two perspectives. Firstly, from the perspective of users, it should provide a useful and easy-access QoS service (i.e., anywhere and anytime QoS service). The QoS allocation must be done in an on-demand and real-time manner. It must also provide the two service categories such as advanced reservation and immediate reservation services. Secondly, in the network administrator's perspective, the QoS allocation system should attempt to provide an integrated resource allocation. Linking with NMS (network management system) functions is also required including the support of graphical network topology and management functions. The desired implementation should exhibit the following characteristics. It must

- Enable on-demand real-time resource allocation.
- Provide an immediate and an advance resource reservation.
- Support multi-hop resource reservation from ingress to egress routers.
- Support automatic collection of routing information.
- Handle resource negotiation between users and QoS allocation system.
- Perform time scheduling of call requests.
- Display network status and traffic monitoring.
- Support java based user interfaces.

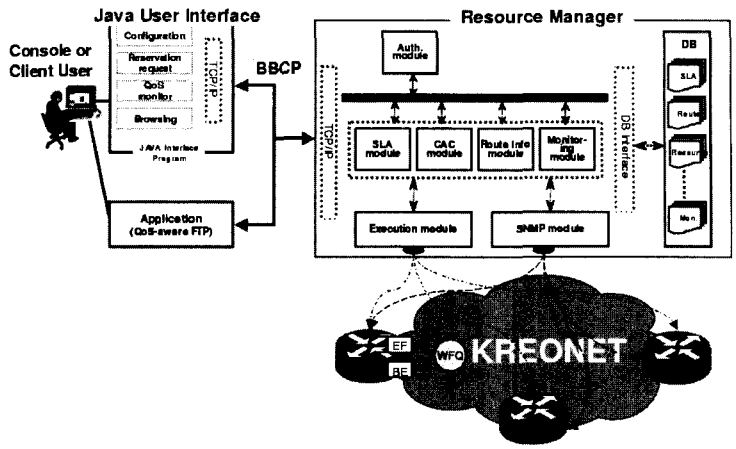


Figure 1: The implementation of QoS allocation system structure.

The rest of the paper is organized as follows. Section 2 describes the basic architecture behind the proposed BB implementation. Call admission method adopted in the BB implementation is introduced in Section 3. Section 4 provides the experimented results, where the implemented system is tested and verified over the supercomputer user network and the access grid testbed in the KREONET/KOREN. Finally, we conclude the paper in Section 5.

II. QoS Allocation System Architecture

One of the objectives in providing QoS to the current best-effort IP network is to enable guaranteed high-speed data transmission for applications and users. The implementation of the proposed QoS allocation system is based on bandwidth management over the DiffServ-enabled network, where commercial routers are configured for its premium service support. In other words, the system is developed to dynamically establish virtual leased line (VLL) in a response to the customer's reservation request.

1. Basic QoS Allocation System Structure

The QoS allocation and resource management system (QoS-ARMS) consists of a resource manager (RM) that allocates and manages network resource, a Java user interface (JUI) that provides the user with an interface to request the network resources, and a set of routers in an administrative DiffServ domain as shown in Fig.1.

The JUI module is simply a JAVA applet that can be downloaded and run in any web browser in Windows or Linux. The user interface provides several management interfaces such as user management, traffic monitoring, router management, resource allocation management, and SLA management. The RM module runs as a server program that provides SLA management module, call admission control (CAC) module, router control module, user authentication module, database management module, and monitoring module. The SLA module manages user's SLA described following:

User ID A unique user ID among existing users maintained by the RM server.

SLA start and end dates The start and end date when the SLA comes into effect and expires.

Source and destination IP address The IP address where service request is invoked (related to the DiffServ edge routers).

Traffic specification For preferable treatment of packets injected to DiffServ edge router, there are traffic description parameters. The edge router is monitoring the injected traffic according to these parameters. If the incoming traffic is violating an agreed rate, the edge routers drop or reshape the violated traffic, and remark the DSCP (DiffServ

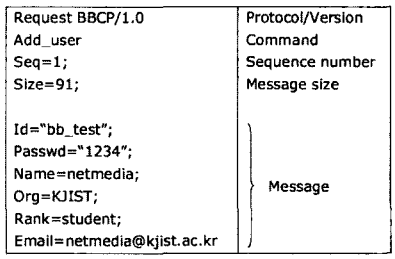
code point) of the packets. The traffic parameters are described following:

- CAB: Conformed average bit rate (Mbps)
- CBR: Conformed burst rate (Mbps)
- CER: Conformed excess rate (Mbps)
- Violation treatments: Droping or shaping

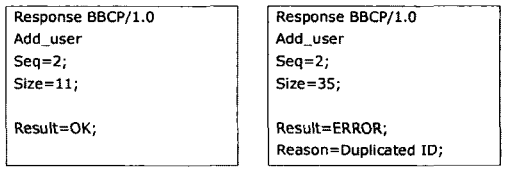
The description of each functional module in the RM is briefly explained as follows. The CAC module processes user's resource request according to the user's SLA. It determines call admission by observing the available bandwidth of routers along the routing path of user's traffic. The router control module collects the routing information, available link bandwidth, and traffic usage of all routers in the administrative DiffServ domain. The router control module updates flow information at the edge routers whenever a user's request is admitted. However, it configures the bandwidth pool of each service class of the core routers only once at the beginning. The user authentication module manages user subscription and login/logout processing. The database management module provides an interface to access the database system. The database table consists of user, SLA, admitted flows, reservations, routers, interfaces of router, monitoring, routing, resources, and login information. The monitoring module periodically observes the routers for current status and provides traffic statistics graphically in real-time. To provide the differentiated service, the class-based waited fair queueing (CB-WFQ) is used in the core routers of the administrative domain.

2. Bandwidth Broker Control Protocol

The *Bandwidth Broker Control Protocol* (BBCP) is a communication protocol between the RM server and the JUI running on the TCP/IP stack. It is a light-weight and text based protocol similar to



(a) Request message



(b) Response message with success (c) Response message with error

Figure 2. Example of BBCP (Adding a new user to the RM server).

HTTP. The BBCP protocol is based on a request/response paradigm. A JUI client establishes a TCP/IP connection with the server and sends a request message consisting of a protocol version, a request headers, and a command type. The server acknowledges the message and sends back reply which contains the protocol version, the status line, a success or error code, and an optional body

Table 1. BBCP commands.

Type	BBCP commands
User information	add_user, user_remove, login, logout
SLA information	sla_add, sla_search, sla_remove
Resource allocation	rar, conform_rar, rar_search, rar_remove
Router information	router_info, get_intf, get_if_edge_core, set_if_edge_core
Monitoring information	topology, get_monitor_info, monitor_if_search
DiffServ configuration	init_diffserv, remove_diffserv

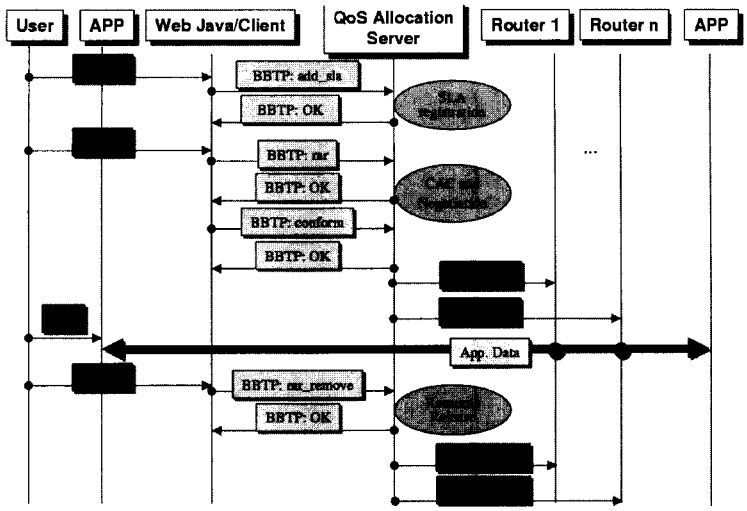


Figure 3. Resource allocation and negotiation procedure.

content. The communication is initiated by a JUI client. The RM server may be engaged in multiple, simultaneous communications with more than one client.

The commands provide a number of interfaces for system operators as well as customers to interact with the RM server and the database. Each command runs as a separate process, taking parameters from the user. The JUI performs basic input validation, checking to see if the data items in the request are valid. It then issues request to the RM server. After processing the request, the RM server sets an exit code on reply which indicates whether the command was successful or not. Table 1 shows several types of BBCP commands, and Fig. 2. exhibits various examples of BBCP.

3. Resource Allocation and Negotiation Procedure

Fig.3 shows the resource allocation procedure in the QoS-ARMS. Initially, users must register their SLA information to the RM server. The QoS allocation request procedure is initiated by sending the *resource allocation request* (rar) message to the RM server from a JUI client. The client determines service rate and flow characteristics based on its SLA. After receiving client's request, the RM server queries the SLA DB (database) to check the validation of the client request and performs the CAC. The CAC result is sent as a response message to the client. If the request cannot be granted, several negotiable information such as current available bandwidth and available time when the request can be accepted are included in the response message. Then, the client sends a confirm message that contains negotiated information: bandwidth and service duration. Finally, the RM server sets up the edge routers to provide QoS, based on the confirmed negotiation. Resource allocation expires when the user explicitly notifies its service termination or when the served time exceeds the agreed service time.

4. Automatic Collection of Router Information in a DiffServ Domain

In the most BB implementations, routing information is collected statically. In this implementation, to collect the routing information automatically, the SNMP (simple network management protocol) tool is used. The RM server accesses the routers and acquires routing information from the router's routing table periodically. Based on the routing information, the RM server can manage the available bandwidth along the path of flows.

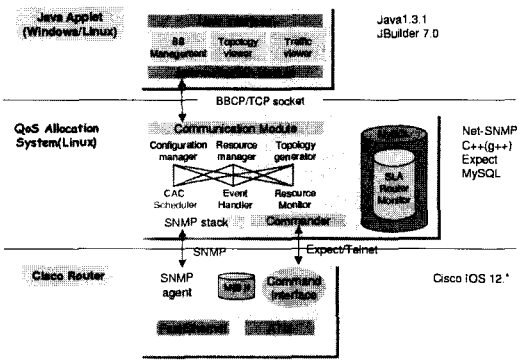


Figure 4. Implementation Environment.

5. QoS Allocation System Implementation and Specification

The QoS allocation system has been implemented using a server/client model. The RM server has been implemented in Linux. Development has been done in C/C++ and *Expect/Telnet has been* used to access router and to update router's configuration automatically. For establishing database system, *MySQL* has been used. To collect the router information, *Net-SNMP* library has been used. The routers in the testbed are *Cisco IOS 12.**. To run the implemented user interfaces, Java 1.3.1 library has been used. The whole environment of the QoS allocation and management system is presented in Fig. 4.

III. New Call Admission Control Method

1. Advanced Resource Reservation

To avoid the rejection at a time when a user wants to utilize network QoS resource, the network resource should be reserved 'immediately' or 'in advance'. In general, the advance reservations have to additionally specify their desired service duration while conventional requests (referred to as immediate reservations that specify amount of required resource) do not. In mixed environment of two types of reservation, the resource manager has to balance them to provide efficient resource utilization. And also, it needs to address the confliction problem caused by indefinite service duration (immediate reservation) between reserved resources (Fig. 5).

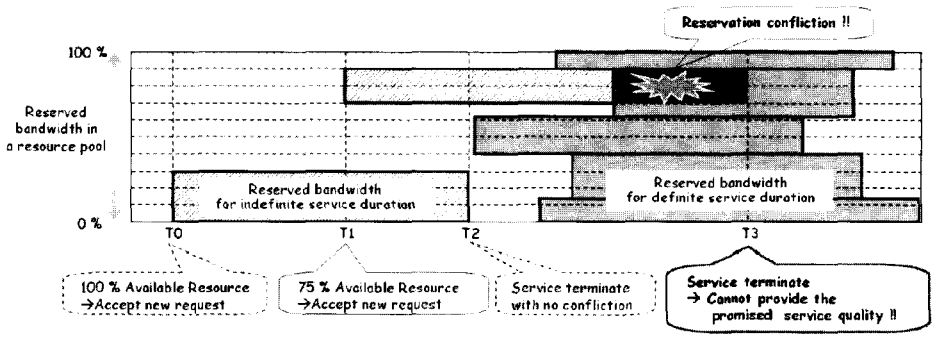
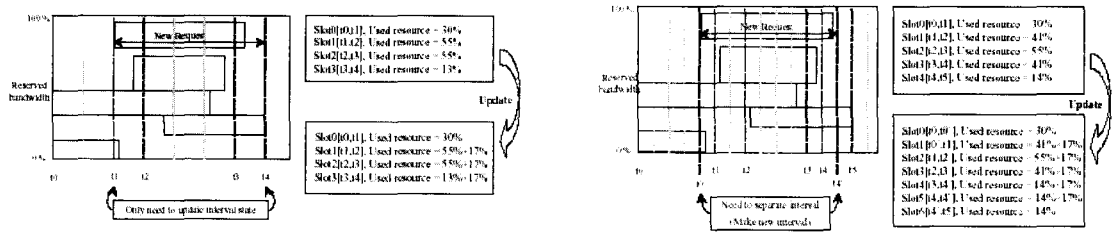


Figure 5. Resource confliction caused by indefinite service duration of immediate reservation.



(a) coarse-grained slot (b) fine-grained slot

Figure 6. Effect on timeslot granularity to the reservation.

To address these issues, we can use the fact that the service durations are partly flexible and can be adjusted through negotiation. However, we need to decide on the time granularity involved in timeslot and enforce different treatment on both immediate and advance reservation types to realize the required reservation feature with the negotiation of service duration. The timeslot table represents the occupation (or allocation) status of whole bandwidth. When a new advance reservation is made, slots are occupied according to start time and duration in the time granularity of timeslot. Thus, the slot size plays an important role in management (Fig. 6).

Fine-grained slot leads to precise resource allocation. However, it also increases processing overhead. Also, to make distinction between advance and immediate reservations, there is a minimum book-ahead time. If the reservation is made before this time, it can be treated as advance one. Note that, the actual amount of minimum book-ahead time depends on the deployment scenario. With this separation, we can handle the reservation requests with different time-domain flexibility and different weight of preemption.

2. Admission control

To make resource reservation, in our implementation, each reservation notifies their requirements through RAR (Resource Allocation Request) message. The RAR message contains reservation specific information as follows.

- SLA_id: Indicating the preferable service level that is selected by user.
- Start_time and End_time: The term of resource usage (in immediate case, there is no end_time)
- Requested_Bandwidth: Amount of preferable resource (this can be changed according to available resource within service duration.

The QoS allocation system builds and manages timeslot table which contains information about resource allocation status in advance. For new reservation request, our system checks the timeslot tables within data path to determine resource availability and updates when the reservation request is accepted. In this work, the time granularity that we use is 1 sec. Also, to handle both types of reservation, we statically divide the resource for each reservation.

In case of the immediate reservation, the RM decides acceptance of new reservation request using Eq. 1. In this process, the resource manager only considers resource availability at that time instance.

$$B_{new} + \sum_{k}^{m} B_k \le B_{\max} \quad 1) \qquad (1)$$

With advance reservation, the resource manager checks Eq. 2 using the timeslot information. If the resource is available within service duration (t_s, t_e), the advance reservation is accepted and the timeslot table is updated.

$$\{B_{new} + \sum_{k}^{m} B_k\}_{t\in\{t_s,t_e\}} \le B_{\max} \quad 2) \qquad (2)$$

Time Negotiation

In this work, the service time is negotiated differently between two types of reservation requests. For the immediate reservation, admission control can only consider the current state of resource availability, since it can not know when the other immediate reservations will terminate. Thus, to avoid unnecessary rejection of immediate reservation requests when there is short-time temporary bottleneck, the admission control just delays the allocation when the bandwidth is not sufficient. To handle this, our admission control first checks the available bandwidth at that time instance. If resource is not sufficient, the request

1) B_{new}: Requested Bandwidth for new reservation, B_k: Requested Bandwidth for kth reservation, B_{max}: Total bandwidth.

2) t_s: Service start time, t_e: Service end time

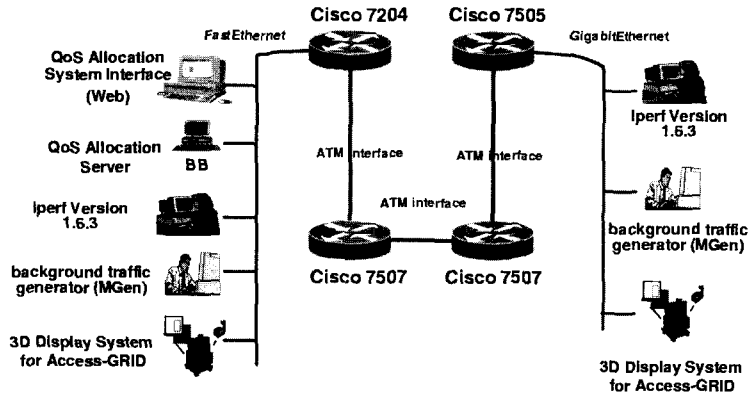


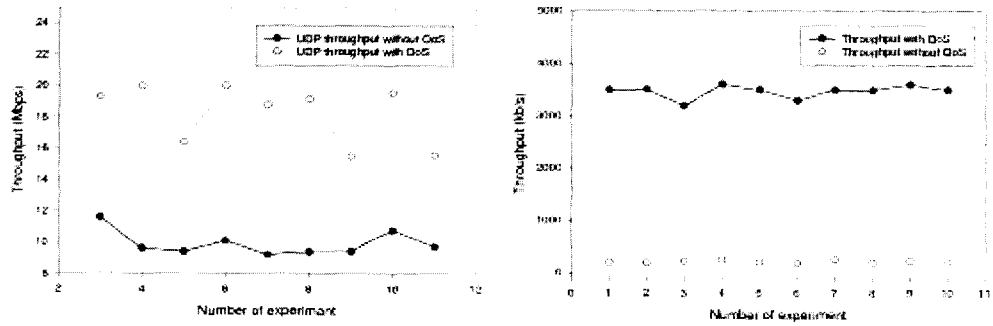
Figure 7. Testbed for the verification of QoS allocation system

is forwarded to the waiting queue. Then, it waits there and periodically re-checked until the user utility related with its flexible service start time becomes zero. To help requests waiting for long in the scheduling queue, those requests receive more weight. With this, each immediate reservation request is sorted by its assigned weight and receives requested resource.

When an advance reservation request is given with the specified service duration, the admission control performs the task of request coordination in order to maximize the resource utilization and aggregated user satisfaction with the constraints in service time flexibility and available resource amount. It first checks the bandwidth availability within request's service duration in advance and starts negotiation of service time using user utility within its flexible range. During several phases of negotiations, if the requested resource is available, the request is accepted and the time table is updated. If the resource is not sufficient to accommodate the request even after it is maximally modified, the request is rejected.

IV. QoS Testbed Configuration and Experiment Result

In this section, we describe the overall framework of the practical experiment over the testbed connected with QoS-ARMS. We consider two kind of QoS services in the experiment: premium service and best effort service. The topology we configured is shown in Fig.7. The testbed of QoS allocation system is established with four CISCO routers over the public KREONET and KOREN. The QOS-ARMS establishes the router 7204 and 7505 as DiffServ edge interfaces and ATM interfaces of routers 7507 as DiffServ core



(a) UDP throughput (b) TCP throughput

Figure 8. Throughput experiments.

Job class	Average processing time	Standard deviation
Database processing	0.967ms	0.588ms
BBCP command processing	80ms	186ms
SNMP query processing	504ms	864ms
QoS setup for edge router	378ms	361ms
Call admission control	275ms	109ms

Table 2. Response time of the resource manager.

interfaces. The RM server is located in the subnet of the router 7204. Subnet connected with each router is FastEthernet that can support up to 100Mbps. We statically set the EF (expedited forwarding) bandwidth through the core interfaces to 25Mbps. The weighted fair queueing (WFQ) is used to provide the EF service of each core and edge routers. A required bandwidth of the EF class is used as a weight of the WFQ in the CISCO's core routers. To generate, measure, and analyze traffic, MGEN and IPerf tools are used. While the *MGEN* generats the background traffic and the *IPerf* tests TCP and UDP traffic, a 3D-stereo video transmission system is injected from both end-sides to measure the performance of the implemented QoS allocation system. During the experiments, the packet size of UDP and TCP traffic generated by the MGEN and the IPerf is 1Kbytes with constant bit rate.

The performance of UDP and TCP transmission is depicted in Fig.8, where background traffic is 20Mbps generated by the MGEN, and 20Mbps is reserved. Thus, the 20Mbps of UDP performance can be achieved by this QoS reservation. However, the UDP performance is limited to 10Mbps without QoS reservation as shown in Fig. 8(a). In Fig. 8(b), the TCP performance is represented. This time 5Mbps QoS is reserved and 25Mbps background traffic is forced into the network. The TCP performance reaches 4Mbps with requesting QoS allocation. However, only 1Mbps is achieved without QoS allocation. Note that the TCP congestion control causes this discrepancy. Fig. 9 denotes measurement results of the RTT variation with and without QoS reservation. When the QoS is allocated, the RTT value drops to 110ms. Without the QoS reservation, the RTT value jumps to 160ms.

Table 2 represents the average and variation of job response times. Jobs that directly control the routers are SNMP processing and QoS setup processing. These jobs take most of the processing time while the jobs to query DB and BBCP message processing need minor processing time. The CAC processing also takes longer time to find routing the path and to lookup available bandwidth of each interface along the routing path.

Fig. 10 represents the 3D-stereo video transmission system linked with Access Grid multi-site next-generation conference tool. The 3D-stereo video requires double the bandwidth of

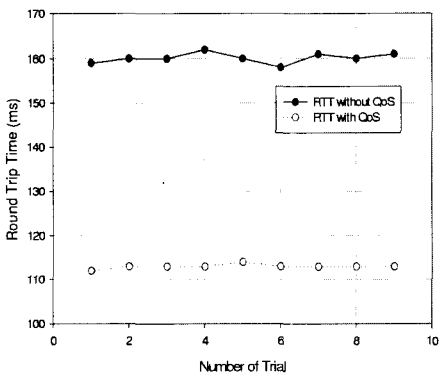


Figure 9. Round trip time.

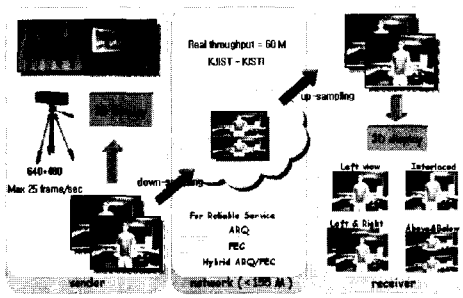


Figure 10. 3D-Stereo video transmission system.

a non-stereo video, since a frame of stereo video consists of two images: one for the left eye and the other for the right eye. The real-time delivery of the 3D-stereo video transmission is a very important requirement for better user collaboration in the Access Grid system. With the QoS reservation, the 3D-stereo video can provide seamless moving images without playback interruption from the background traffic.

International QoS Test through EMERGE-2 QoS Testbed

ESnet/MREN regional grid experimental NGI testbed-2 (EMERGE-2) is a joint project between EVL/UIC, NWU and KISTI. The Objective of EMERGE-2 is to tune the DiffServ network in order to satisfy the performance requirements of the high bandwidth applications (i.e., CAVE and Access Grid). The EMERGE-2 test includes EMERGE-2 testbed connectivity testing through STARTAP, deployment of new version2 resource manager&verification (GARA-based DiffServ Manager). It also addresses TCP adaptation, differentiated transmission of MPEG-2 video, and a reliable blast UDP (RBUDP) adaptation over the DiffServ network. The proposed QoS testbed is connected to EVL-NU QoS testbed via STARTAP as shown in Fig. 11. EMERGE-2 is currently undergoing testing.

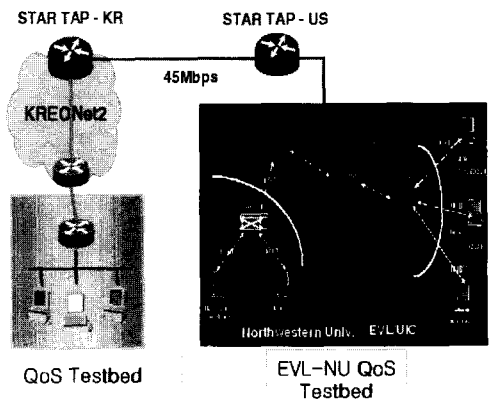


Figure 11. EMERGE-2 testbed configuration.

V. Conclusion

In this paper, we have described the implementation of a Bandwidth Broker system that performs in a separately administrated DiffServ domain to support QoS to various media applications. The implemented system provides real-time resource allocation, available resource analysis, resource reservation, enhanced call admission control, QoS negotiation between user and server, and resource observation over the single administrative domain. The implemented system can contribute for establishing policy-based QoS allocation system for the next generation Internet.

References

[1] S. Black, et. al., "An architecture for differentiated services," IETF RFC 2475, 1998.

[2] G. Hwang, D. Lee, J. Jo, J. Kim, S. Jung, and O. Byeon, "Implementation of an on-demand bandwidth-based QoS allocation system for end-to-end resource reservation," in *Proc. JCCI'01*, 2001.

[3] B. Teitelbaum, S. Hares, L. Dunn, R. Neilson, V. Narayan, and F. Reichmeyer, "Internet2 QBone: building a testbed for differentiated services," *IEEE Network*, vol. 13 Issue 5, Page 8-16, Sep 1999.

[4] I. Khalil and T. Braun, "Implementation of a bandwidth broker for dynamic end-to-end resource reservation in outsourced virtual private networks," in *Proc. LCN'02*, Page 511-519, 2000.

[5] A. Terzis, L. Wang, J. Ogawa, and L. Zhang, "A two-tier resource management model for the Internet," in *Proc. GLOBECOM'99*, vol. 3, Dec. 1999.

[6] Z. Zhi-Li, D. Zhenhai, and Y. T. Hou, "On scalable network resource management using bandwidth brokers," in *Proc. IEEE/IFTP NOMS'02*, 2002.

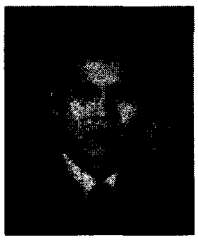
[7] Z. Zhi-Li, D. Zhenhai, L. Gao, and Y. T.

Hou, "Decoupling QoS control from core routers: A novel bandwidth broker architecture for scalable support of guaranteed services," in *Proc. ACM SIGCOMM'00*, August 2000.

[8] M. Hashmani and M. Yoshida, "ENICOM's bandwidth broker," in *Proc. Symposium on Applications and the Internet Workshops*, 2001.

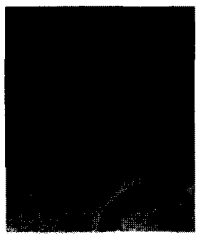
[9] S. Jha and M. Hassan, "Implementing bandwidth broker using COPS-PR in Java," in *Proc. LCN'01*, 2001.

이 동 욱 (Dongwook Lee) 정회원



1997년 2월 : 충북대학교 컴퓨터공학과 학사
1999년 2월 : 광주과학기술원 정보통신공학과 석사
2004년 2월 : 광주과학기술원 정보통신공학과 박사
2004년 3월~현재: 광주과학기술원 정보통신공학과 연구원
<관심분야> Mobile Network, Network- adaptive Media Delivery, Performance Measurement, Multicast, and End-to-end QoS

김 종 원 (JongWon Kim) 정회원

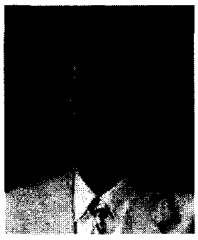


1987년 2월: 서울대학교 제어계측공학과 학사
1989년 2월: 서울대학교 제어계측공학과 석사
1994년 2월: 서울대학교 제어계측공학과 박사
2001년 9월~현재: 광주과학기술원 정보통신공학과 부교수
2000년 7월~2001년 6월: 미국 InterVideo Inc., Fremont, CA, 개발자문
1998년 12월~2001년 7월: 미국 Univ. of Southern California, Los Angeles, CA, EE-Systems Department 연구조교수
1994년 3월~1999년 7월: 공주대학교 전자공학과 조교수
<주관심분야> Networked Media Systems and Protocols focusing "Reliable and Flexible Delivery for Integrated Media over Wired/Wirelss Networks
(네트워크미디어:http://netmedia.kjist.ac.kr)

이 동 훈 (DongHoon Yi) 준회원

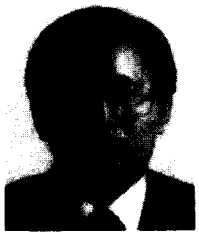
2001년 2월. 인하대학교 전자공학과 학사)
2003년 2월. 광주과학기술원 정보통신공학과 석사
2003년 3월 ~ 현재. 광주과학기술원 정보통신공학과 네트워크미디어 연구실 박사 과정
<주관심분야> Internet QoS, Network Resource Management, Call Admission Control. Network Calculus

정 상 길 (Sangkil Jung) 정회원



1999년 2월: 충남대학교 통계학과 학사
2001년 8월: 광주과학기술원 정보통신공학과 석사
2003년 9월~현재: State University of New York at Stony Brook, Computer Science 박사과정
2001년 9월~2003년 7월: 한국과학기술원 인프라개발실 연구원
<주관심분야> Development of analytical methodologies for enhancing wired and wireless QoS.

변 옥 환(Okhwan Byeon) 정회원



1979년, 한국항공대학교 통신정보공학과
1993년, 경희대학교 전자공학과 (박사)
고려대학교, 경희대학교 겸임교수 역임
미국 OSM Corp(1984), NCSA/UIUC(1997) 객원연구원
1978년-1995년 KIST 시스템공학연구소 책임연구원, 연구전산망개발실장, 슈퍼컴퓨팅응용실장
1995년-1999년 ETRI 슈퍼컴퓨팅센터 책임연구원, 고성능망연구실장, 슈퍼컴퓨팅연구실장
1999년-현재 KISTI 슈퍼컴퓨팅센터 책임연구원, 초고속연구망부장, 슈퍼컴퓨팅인프라개발실장
<주관심분야> 고성능망 관리 및 보안, 그리드네트워킹 및 협업액세스그리드, 유비쿼터스 네트워킹 및 모바일 그리드