

# IP QoS의 확장성을 위한 트래픽 관리 방안

## (A Traffic Management Scheme for the Scalability of IP QoS)

민 안 기<sup>\*</sup> 석 정 봉<sup>\*\*</sup>  
(An-Ki Min) (Jung-Bong Suk)

**요 약** IETF에서는 높은 대역폭과 낮은 지연시간을 요구하는 음성 및 영상 스트림을 포함한 새로운 종류의 응용 서비스의 QoS를 지원하기 위한 Intserv와 RSVP를 정의하고 있다. 그러나, 현재의 Intserv 모델에서는 각 노드가 각 flow의 상태를 유지해야 하므로, 망의 규모가 커질수록 노드의 구조가 복잡해지고 패킷의 처리 속도도 저하되는 문제점을 안고 있다. 본 논문에서는 이러한 stateful 망 구조의 확장성 문제를 극복하기 위해, core 노드에서 각 flow 상태를 유지하지 않고 edge 노드에서만 각 flow 상태를 유지하도록 제안된 SCORE(Scalable Core) 네트워크 구조를 Intserv의 각 서비스 QoS를 만족시킬 수 있도록 확장하였으며, 이를 위한 수락제어, 대역폭 할당 방식 및 노드 구조를 제안하였다. 또한, 각 flow에 대한 대역폭 할당, 패킷 지연 및 지연시간의 변이를 성능 변수로 하여, 제안 방식에 대한 성능 실험을 ns-2 시뮬레이터를 이용하여 수행하였으며, 이를 통해 제안 방식이 Intserv에서 제시한 각 서비스의 서비스 품질 요구 사항을 충분히 만족시키면서 Intserv 모델의 단점인 확장성을 문제를 해결할 수 있는 좋은 방안임을 확인하였다.

**키워드** : 인터넷, 서비스 품질, 통합 모델, 스케줄링, ns-2 모의 실험

**Abstract** The IETF has defined the Intserv model and the RSVP signaling protocol to improve QoS capability for a set of newly emerging services including voice and video streams that require high transmission bandwidth and low delay. However, since the current Intserv model requires each router to maintain the states of each service flow, the complexity and the overhead for processing packets in each router drastically increase as the size of the network increases, giving rise to the scalability problem. This motivates our work; namely, we investigate and devise new control schemes to enhance the scalability of the Intesev model. To do this, we basically resort to the SCORE network model, extend it to fairly well adapt to the three services presented in the Intserv model, and devise schemes of the QoS scheduling, the admission control, and the edge and core node architectures. We also carry out the computer simulation by using ns-2 simulator to examine the performance of the proposed scheme in respects of the bandwidth allocation capability, the packet delay, and the packet delay variation. The results show that the proposed scheme meets the QoS requirements of the respective three services of Intserv model, thus we conclude that the proposed scheme enhances the scalability, while keeping the efficiency of the current Intserv model.

**Key words** : Internet, QoS, Intserv, Scheduling, ns-2 Simulation

### 1. 서론

현재의 인터넷은 최선형(best-effort) 서비스만을 지원하고 있고, 이러한 인터넷의 최선형 서비스 환경에서는

응용 프로그램의 안정성이 전송에 따른 지연보다도 중요했으며, 데이터 전송 시에 어느 정도의 지연은 서비스 특성상 그리 큰 문제가 되지 않았다. 그러나 인터넷 전화, 인터넷 방송, 멀티미디어 서비스 등의 실시간 혹은 높은 대역폭을 요구하는 서비스들이 늘어남에 따라 인터넷에서도 서비스 품질(Quality of Service: QoS)에 대한 요구가 문제가 되고 있다. 따라서, 기존의 최선형 서비스뿐만 아니라, 실시간 서비스 등의 다양한 형태의 정보를 수용하기 위하여 각 서비스에서 요구하는 서비스 품질을 만족

<sup>\*</sup> 비 회 원 : (주)씨앤드씨 엔지니어링 연구원  
amin@pairlink.com

<sup>\*\*</sup> 통신회원 : 연세대학교 전자계산학과 교수  
jbsuk@dragon.yonsei.ac.kr

논문접수 : 2001년 8월 27일

심사완료 : 2002년 5월 2일

시킬 수 있는 방안이 필요하게 되었다.

인터넷에서 실시간 통신 서비스를 위한 프로토콜은 Internet Engineering Task Force(IETF)를 중심으로 정립되고 있는데, 신규 서비스 및 트래픽 파라미터 등을 정의하는 Integrated Services(Intserv) 작업 그룹, 서비스 그룹별 차등화 서비스를 정의하는 Differentiated Services(Diffserv) 작업 그룹, 구체적인 하부 망에 적용하기 위한 기술을 개발하는 Integrated Services over Specific Link Layers(ISSLL) 작업 그룹, 실시간 서비스를 위한 예약 프로토콜을 개발하고 있는 Resource Reservation Protocol(RSVP) 작업 그룹 등의 여러 작업 그룹에서 집중적으로 논의되고 개발되고 있다. 이밖에 QoS에 기반 하여 라우팅하기 위한 기술을 연구하는 QoS 라우팅 작업 그룹, 실시간 서비스를 위한 트랜스포트 계층 이상을 다루는 Audio Video Transport(AVT), Multiparty Multimedia Session Control(MMUSIC) 작업 그룹 및 기존 트래픽을 ATM 등의 망에 전달하고자 하는 ION 작업 그룹 등에서도 이루어지고 있다. 또한 IEEE에서 switched LAN에서 실시간 서비스를 위한 트래픽 클래스등을 정의한 801.1d(p)와 ATM 포럼의 LAN emulation, Multi-Protocol Over ATM(MPOA) 등도 연관되어 있다.

인터넷 통합 서비스 모델을 연구하는 Intserv 작업 그룹[1]에서는 기존의 최선형 서비스(BE: Best Effort) 외에 보장형 서비스(GS: Guaranteed Service) 및 부하제어형 서비스(CLS: Controlled-Load Service)를 정의하고 있다[2, 3]. 이러한 서비스를 실현하기 위해서는 기존의 IP 프로토콜 이외에 망 내에서 일정한 자원을 예약하기 위한 신호 프로토콜인 RSVP[4]가 필요하며, 노드는 각 flow의 상태 관리 및 각 flow의 패킷 처리 및 동작을 수행해야 한다. 각 flow의 상태 정보를 저장하고, 이를 이용하여 도착 패킷에 대한 처리를 하는 이와 같은 stateful 네트워크 구조는 flow 당 서비스를 관리하기 때문에 QoS 지원을 효율적으로 할 수 있으나, 망의 규모가 커질수록 노드의 구조가 복잡해지고, 패킷 처리 시간도 증가하기 때문에 core 망에 적용하기가 어려워지는 확장성(scalability)의 문제가 발생한다.

I. Stoica는 이러한 확장성의 문제를 극복하기 위하여 SCORE(Scalable CORE)라고 하는 망 구조를 제시하였다[5]. SCORE 망에서는 패킷 헤더에 flow의 상태를 저장하고 있는 동적 패킷 상태(DPS: Dynamic Packet State)라는 개념을 정의하고 있다. Ingress 노드에서는 flow의 상태를 계산하여 이를 패킷 헤더에 위치시키고, 패킷이 망을 거쳐갈 때 core 노드는 이 정보에 따라 패킷

을 처리하고 필요에 따라 DPS 정보와 자신의 상태를 갱신시킨다. Egress 노드에 도착한 패킷은 DPS 정보를 패킷 헤더로부터 제거된 후 목적지 호스트에 패킷을 전달하게 된다. Flow 당 QoS를 지원하기 위하여는 네트워크 내에서 각 flow에 대한 정보가 필요하게 되는데, SCORE 망에서는 이 정보를 DPS 라는 이름으로 패킷 내에 담으려서, Intserv와 같은 기존의 stateful 네트워크에서 노드가 각 flow의 상태를 관리해야 하는 부담과 이에 따른 망 확장성의 문제를 극복하고자 한 것이 기본 개념이다. Stoica는 이러한 SCORE 망을 기반으로 하여 FQ 스케줄링 방식을 이용하여 TCP 및 UDP throughput 성능을 분석하였다[6].

본 논문은 Stoica가 제안한 SCORE 망을 기반으로 하여 Intserv 모델에서 정의한 GS 및 CLS 서비스의 품질에 대한 보장이 제대로 달성되는지를 알아보기 위해, 서비스별 대역폭 할당, 패킷 지연 및 지연시간 변이를 성능 변수로 하여 시뮬레이션 실험을 수행하였다. 본 논문은 기본적으로 Stoica가 제안한 SCORE 망 모델을 기반으로 하고 있으나, 다음과 같은 관점에서 기존의 Stoica 연구[5, 6]에 대한 확장 및 차별화 내용을 다루고 있다. 첫째, 참조문헌[5]에서는 사용자 트래픽에 대해 서비스 구분을 하고 있지 않는 단일 서비스 클래스를 대상으로 하고 있으나, 본 논문은 Intserv 모델에서 고려하고 있는 GS, CLS 및 BE의 세 가지 서비스 클래스를 고려하였고, 이에 따른 각 서비스의 패킷 처리를 위한 edge 및 core 노드의 구조, 대역 자원 할당 방법이 고려되었다. 둘째, 패킷의 스케줄링 방안으로는 Generalized Process Sharing(GPS)에 가장 유사한 성능을 보임으로서, 각 서비스에 대한 대역폭의 보장과 최소한의 지연시간의 보장을 효과적으로 할 수 있는 것으로 알려진 WF2Q 방식을 사용하였다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서 본 연구와 연관성이 있는 QoS를 지원하기 위한 기존의 연구의 내용을 다룬다. IETF에서 정의한 두 가지 서비스 모델인 Intserv 및 Diffserv, 그리고 Stoica가 제안하고 본 연구의 대상인 SCORE 망의 기본 동작과 Intserv와 같이 서비스 구분에 따른 패킷 스케줄링 방식에 대하여 설명한다. 3장에서는 Stoica가 제안한 단일 서비스 SCORE 망 기반으로 하여 이를 Intserv의 세 가지 서비스 클래스로 확장한 SCORE 망에서의 수락제어, WF2Q 기반의 QoS 스케줄링을 이용한 대역 할당 방법, 그리고 edge 및 core 노드 구조를 제안한다. 4장에서는 제안 방식이 Intserv 모델의 각 서비스별 품질에 대한 보장이 제대로 달성되는지를 알아보기 위해, 서비스별 대역폭 할당, 패킷 지연 및

표 1 스케줄링 알고리즘 비교

스케줄링 알고리즘	개요	특징
Fair Queueing	<ul style="list-style-type: none"> <li>• flow별 라운드 로빈 방식 서비스</li> <li>• 모든 flow에 대한 대역폭과 패킷 길이가 동일시에만 의미 있음</li> </ul>	<ul style="list-style-type: none"> <li>• 입력마다 분리된 버퍼 사용</li> <li>• 각 흐름에 할당된 대역폭이 서로 다르고 패킷의 길이가 다른 환경에서 부적합</li> </ul>
WFQ (Weighted Fair Queueing)	<ul style="list-style-type: none"> <li>• 입력 패킷에 서비스 순서를 표시하는 time stamp를 찍어 스케줄링</li> <li>• 입력 트래픽의 양에 따른 virtual time 값을 이용해 새로이 입력되는 패킷에 대한 time stamp를 계산하여 입력 패킷에 첨부</li> <li>• 가상 종결시간이 가장 작은 패킷을 다음에 먼저 전송</li> </ul>	<ul style="list-style-type: none"> <li>• 대역폭 보장</li> <li>• leaky-bucket 제어된 입력 트래픽에 대해 지연시간 보장</li> <li>• 고속망에서는 구현 힘들(가상 종결시간 계산을 위한 GPS 시뮬레이션을 해야 함)</li> </ul>
WF <sup>2</sup> Q (Worst-case Fair Weighted Fair Queueing)	<ul style="list-style-type: none"> <li>• 보다 정확한 GPS 방식 시뮬레이션을 위해 시작시간과 종료시간을 사용하여 서비스 순서 결정</li> <li>• GPS 방식으로 서비스할 때 시간 t에서 서비스를 요구하는 패킷들만의 가상종결시간을 계산하여 가장 작은 값을 갖는 패킷을 우선 전송</li> </ul>	<ul style="list-style-type: none"> <li>• GPS와 유사한 성능 발휘</li> <li>• 최소한의 지연시간 보장</li> <li>• 고속망에서는 구현이 힘들(GPS 시뮬레이션을 통해 가상 종결시간 및 가상 시작시간 계산)</li> </ul>

지연시간 변이를 성능 변수로 한 모의실험 및 성능분석 결과를 다루고, 마지막으로 5장에서 결론을 맺도록 한다.

## 2. QoS 지원을 위한 기존 연구

### 2.1 IETF 서비스 모델

IETF에서는 인터넷 상에서 실시간형 서비스의 QoS를 지원하기 위한 방안으로 Intserv[1]와 Diffserv[5] 모델을 정의하고 있는데, 서비스의 정의와 구현 기술에 있어서 차이점을 가지고 있다. 먼저, 서비스의 정의에 있어서 Intserv는 각 flow에 기초한 종단 간(end-to-end) 보장형 서비스와 부하제어형 서비스를 제공하고 있는 반면, Diffserv는 현재의 인터넷에 대규모의 수정을 가하지 않고 간단한 방법으로 각 데이터 패킷의 식별 기능을 추가하여 서비스의 차별화를 제공하고 있다. 또한, 구현 기술에 있어 현재 Intserv 방법은 각 노드가 각 flow의 신호 메시지를 처리하며 경로 상에 각 flow의 데이터 전송과 QoS 상태를 유지하고 각 flow의 분류, 스케줄링, 버퍼 관리를 수행하는 반면, Diffserv는 edge 노드와 core 노드를 구별하여, edge 노드는 per flow, per organization 과 같은 트래픽 단위에 기초해서 패킷을 처리하는 반면 core 노드는 각 flow의 상태 정보를 유지하지 않고 패킷 헤더의 bit 패턴에 의해 표시되어 있는 DSCP(DS codepoint) 값에 근거해서 패킷을 처리한다.

이 두 서비스 모델의 장단점을 살펴보면, 먼저 Intserv는 네트워크 내에 각 노드에서 예약 상태를 설정하고 유지하는 각 flow의 상태 관리를 하기 때문에 각 flow의 종단 간 QoS가 보장되고 높은 효율을 가질 수 있는 반면, flow의 수가 증가함에 따라 per flow 동작의 복잡성이 증가하기 때문에 확장성에 문제를 가진다. Diffserv는 edge 노드에서 복잡한 처리를 수행하고

core 노드는 상대적으로 간단하기 때문에 망의 확장에 용이하나, 각 flow의 상태 관리를 하지 않기 때문에 노드의 효율성이 떨어질 수 있으며, 이러한 문제를 보완하기 위해 수락제어의 효율성을 높이는 bandwidth broker 등의 방법들이 활발히 연구되고 있다.

### 2.2 Stateful 네트워크의 스케줄링 방식

서비스 분류가 되고, 이에 따라 노드에서 flow 당 패킷 처리를 달리 하는 stateful 네트워크에서는, 각 flow의 QoS를 만족시키기 위한 효과적인 스케줄링 방식이 매우 중요하다. Intserv와 같은 stateful 네트워크에서 많이 사용되는 스케줄링 방식으로는 Fair Queueing(FQ)[8], Weighted Fair Queueing(WFQ)[9], Worst-case Fair Weighted Fair Queueing(WF2Q)[10] 등이 있는데, 각 방식에 대한 주요 동작 및 특징은 [표 1]과 같다.

[그림 1]은 노드에 연결된 flow의 패킷에 대한 WFQ와 WF2Q 스케줄링 방식의 서비스 순서를 보여주고 있다. 특히, WF2Q의 패킷 서비스 순서는 임의 시점에서 서비스를 요구하는 패킷들만의 가상종결시간을 계산하여 가장 작은 값을 갖는 패킷을 우선 전송함으로써 각

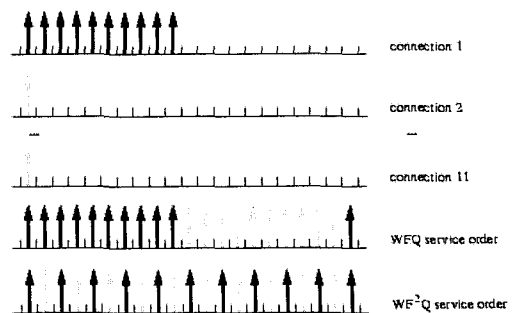


그림 1 WF2Q Service Order

flow에 대해 최소한의 지연시간을 보장할 수 있게 한다.

### 2.3 SCORE 네트워크

I. Stoica는 Intserv의 확장성 문제를 해결 위한 방안으로, [그림 2]와 같은 SCORE 네트워크 구조 제안하였다[5]. SCORE 네트워크의 목적은 RSVP를 사용하는 Intserv 네트워크와 같은 stateful 네트워크에서 제공되는 서비스와 유사한 서비스를 제공하고자 함에 있었고, 이를 만족시키기 위해 [그림 3]의 Dynamic Packet State(DPS) 개념을 도입하였다. SCORE 네트워크에서는 Diffserv와 유사하게, 노드를 edge 노드와 core 노드로 구분하는데, 각 flow 상태의 관리는 edge 노드만이 간여함으로써 core 노드 구조가 간단하게 된다. Edge 노드에서 초기화되는 각 flow의 상태 정보인 DPS는 패킷 헤더에 실려 망 내부로 운반되고, core 노드들은 다음 hop으로 패킷을 전송하기 전에 내부 상태와 패킷 헤더의 DPS 정보를 갱신한다. 따라서, edge 노드에서만 각 flow의 상태 정보를 유지하고 core 노드에서는 패킷 헤더의 상태 정보를 얻어서 패킷을 처리하게 되므로 그 구조가 간단하게 되지만, DPS 내에 있는 각 flow의 상태 정보를 이용하여 패킷을 처리하기 때문에 각 flow당 관리를 하는 stateful 네트워크와 같은 효과를 얻을 수 있다. Stoica는 SCORE 네트워크를 기반으로 하여 수락제어 방안 및 Fair Queueing(FQ) 스케줄링을 이용한 대역 할당 방안을 제시하고 있다[6].

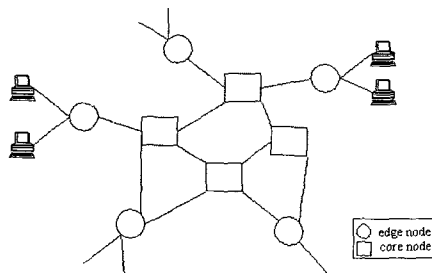


그림 2 SCORE 네트워크 구조

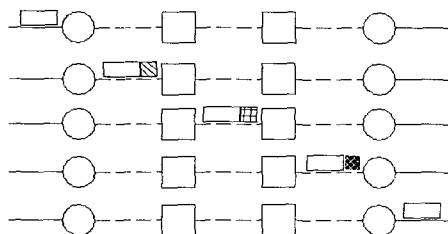


그림 3 동적인 패킷 상태 정보 갱신

새로운 flow에 대한 수락은 새로운 flow의 전송 예약율이 출력 링크 용량과 현재 서비스 중인 전체 flow의 전송 예약율 차이보다 작을 경우에 이루어진다. FQ 스케줄링에 의한 대역폭 할당은, 먼저 노드에서 각 flow의 상태 정보를 기반으로 하여 공평율을 예측(estimated fair rate  $f$ )하여, 이 값과 패킷 도착율  $r$  중 적은 값을 택함으로써 이루어진다. 여기서, 공평 예측율  $f$ 는 각 flow에 할당된 대역폭 합이 출력 링크 용량보다 적게 되도록 계산된다.

### 3. SCORE 구조를 확장한 제안 방식

본 연구는 Stoica가 제안한 단일 클래스 SCORE 망을 Intserv에서 제시한 세 가지 서비스 클래스, 즉 GS, CLS 및 BE에 적용될 수 있도록 확장시켰으며, 이에 따른 각 서비스의 QoS가 지원될 수 있도록 수락제어 방식을 확장 수정하였다. 또한, 패킷 스케줄링 방식으로는 GPS에 가장 유사한 성능을 발휘하는 것으로 알려진 WF2Q 방식을 채택하였으며, 패킷 처리를 위한 edge 및 core 노드의 구조를 제시하였다.

#### 3.1 수락제어

Intserv 모델에서 QoS를 제공하기 위한 중요한 요소의 하나는 수락제어(admission control)이다. 수락제어는 어떤 링크 상에 flow의 예약율 합이 링크의 용량을 초과하지 않도록 새로운 요청의 수락여부를 결정하는 것으로, 만약 새로운 예약 요청이 경로 상에 각 홉에서 수락 테스트를 통과하면 이 요청은 받아들여진다. Intserv에서 제안된 RSVP를 사용하는 네트워크에서는 이러한 수락제어가 경로 상에 모든 노드에서 수행되므로 모든 flow의 동적인 상태를 관리해야 하기 때문에 확장성의 문제를 안고 있다. 제안된 방식에서는 Intserv 모델에서 제시한 각 서비스가 요구하는 모든 flow의 예약율이 네트워크의 자원을 초과하지 않는 범위에서 수락제어를 하도록 한다. Ingress 노드와 Egress 노드 사이의 수락제어는 [그림 4]와 같이 이루어진다.

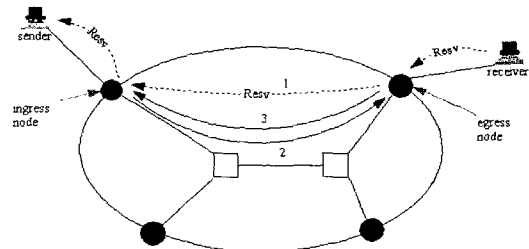


그림 4 SCORE 네트워크의 수락제어

[그림 4]의 SCORE 도메인 내에서는 신호 프로토콜을 사용한다. Edge 노드들은 이 신호 프로토콜과 RSVP 사이의 인터페이스 역할을 한다. Path, Resv와 같은 RSVP의 제어 메시지들은 네트워크의 edge 노드에서만 처리되고 core 노드에서는 아무런 처리되지 않고 통과된다. Ingress 노드에서 Path 메시지를 받았을 때 이 메시지는 단지 내부 도메인을 거쳐 전송되고, egress 노드에 Resv 메시지가 도착했을 때 이것은 내부 노드를 거쳐 ingress 노드로 전송된다. 다음으로 ingress 노드에서 수락제어 신호(메시지 2)를 broadcast 해서 core 노드에 대한 수락제어를 하게 된다. 메시지 2는 core 노드를 거쳐가면서 각 hop 수를 카운트하고 경로를 기록한다. 메시지 2가 egress 노드에 도착했을 때 egress 노드는 hop 카운트가 가장 적은 경로를 선택하고, 메시지 3을 통해서 최종 경로를 ingress 노드에게 알려준다. 이 경로는 ingress 노드에서 패킷의 헤더에 기록되어 source routing 형태로 모든 데이터 패킷은 동일한 경로로 전송된다.

Stoica가 제안한 SCORE 네트워크에서 core 노드의 수락제어는 아래의 식에 준하여 이루어진다[5]. 즉, 예약률  $R$ 은 식 (1)에서와 같이 자신의 링크 용량(link capacity)  $C$ 를 과하지 않도록 유지한다. 예약률  $r$ 을 가진 flow가 링크 용량  $C$ 를 초과하지 않는다면 전체 예약률  $R$ 은 식 (2)에서와  $r$  만큼 증가하고 수락이 이루어진다. 예약률  $r'$ 을 가진 flow가 종료되었을 때 전체 예약률  $R$ 은 (3)에서와 같이  $r'$  만큼 감소한다.

$$R = \sum_i r_i < C \tag{1}$$

$$R = R + r \leq C \tag{2}$$

$$R = R - r' \tag{3}$$

이 방식은 보장형 서비스를 요구하는 flow에 대한 예약과 QoS 보장을 지원하기 때문에 GS 서비스 요구가 많을 경우 각 노드의 자원이 GS 서비스에 집중되는 문제로 인해 CLS와 BE 서비스에 대해 자원을 할당할 수 없는 문제와 노드의 한 링크에 많은 GS 서비스가 할당되었을 때 다른 링크에는 자원을 할당할 수 없는 문제가 발생할 수 있다. 이러한 자원의 집중을 막기 위해서 제안된 방식에서는 각 노드에서 계층적인 자원 관리를 수행하도록 하였다. 즉, 각 노드에는 자원 관리 장치(Resource Manager)를 두고, 이 장치는 각 링크에 할당된 링크 용량 및 현재 각 서비스의 예약율에 대한 정보와 현재 각 서비스 flow의 도착율을 측정하여 각 서비스에 동적으로 자원을 할당하게 된다. 이렇게 각 서비스에 정해진 자원 할당량에 따라 GS와 CLS에 대해 각 예약율을 이용해서 아래와 같이 수락제어를 수행한다.

$$C_{tot} = C_{GS} + C_{CLS} + C_{BE} \tag{4}$$

$$R_{GS} = R_{GS} + r_{GS} \leq C_{GS} \tag{5}$$

$$R_{CLS} = R_{CLS} + r_{CLS} \leq C_{CLS} \tag{6}$$

Core 노드의 전체 링크 용량  $C_{tot}$ 는 (4)에서와 같이 GS 링크 용량  $C_{GS}$ , CLS 링크 용량  $C_{CLS}$ , BE 링크 용량  $C_{BE}$ 로 나뉜다. BE의 경우는 예약이 필요하지 않고 GS와 CLS의 경우 새로운 flow가 허용되었을 때 각 예약율은 (5), (6)에서와 같이 각 서비스에 할당된 링크 용량을 넘지 않는 범위에서 예약이 이루어지도록 한다. 이와 같이 계층적인 자원 할당을 통해서 노드의 자원을 사용해서 자원이 하나의 링크 또는 서비스에 집중되는 것을 막고 자원을 효율적으로 사용할 수 있다.

### 3.2 QoS 스케줄링

스케줄링이란, 서비스 요구사항에 따라 구별된 패킷 흐름들을 개개의 서비스 품질을 만족할 수 있도록 서비스 순서를 결정하는 방법이다. 중요한 QoS 파라미터 중의 하나인 전달지연은 전적으로 망이 제공하는 서비스 실행순서에 의존하기 때문에 패킷 스케줄링은 트래픽 제어의 핵심 요소이다. 그러나, 실질적으로 통신망에는 서로 다른 QoS를 갖는 수많은 패킷들의 흐름으로 구성되며, 링크는 이렇게 생성된 여러 흐름들에 의해 공유된다. 따라서, 각 흐름마다 고유의 서비스 품질을 제공하기 위한 방법으로 매 순간마다 이용 가능한 여러 흐름 중에서 서비스 순서를 결정하는 것은 매우 어려운 문제이다. 또한 망 전체를 거처가는 전달 경로의 모든 패킷들에 대한 QoS의 만족은 지속적으로 적용되어야 하기 때문에 최적화 또한 어려운 문제이다. 연결이 설정되면 각 연결의 입력 패킷들은 서비스의 종류, 즉 실시간 또는 비실시간 서비스에 따라 다른 종류의 흐름 유형을 갖게 되며, 망은 이러한 차이를 구별하여 패킷들마다 차등적인 서비스를 제공해주어야 한다. 이를 위해 다양한 트래픽을 흐름 종류별로 분류된 패킷들은 각 흐름열의 중요도나 각 패킷의 마감시간에 따라 다음에 서비스될 패킷 순서를 결정하고 조정함으로써 QoS를 보장해줄도록 작동해야 한다.

현재 인터넷에서는 FIFO 큐잉만을 제공하고 있으며 이것은 단지 최선형 서비스만을 고려한 것이기 때문에 Intserv에서 제안하고 있는 각 서비스 유형에 따른 종단간 지연을 보장할 수 없다. 이러한 문제점을 해결위하여, 본 제안 방식에서는 GPS와 가장 유사한 성능을 보임으로서, 각 서비스에 대한 대역폭의 보장과 최소한의 지연시간의 보장을 효과적으로 할 수 있는 것으로 알려진 Worst-case Fair Weighted Fair Queueing(WF2Q) 스케줄링 방식을 사용하였다[10]. WF2Q 스케줄링 방식

을 이용한 서비스별 대역 할당은 다음과 같이 이루어진다.

$$W_{tot} = W_{GS} + W_{CLS} + W_{BE} = C_{tot} \quad (7)$$

$$S_{GS} = \min(a_{GS}, W_{GS}) \quad (8)$$

$$S_{CLS} = \min(a_{CLS}, W_{CLS}) \quad (9)$$

$$S_{BE} = \min(a_{BE}, W_{BE}) \quad (10)$$

$$S_{tot} = S_{GS} + S_{CLS} + S_{BE} \leq C_{tot} \quad (11)$$

먼저, 각 서비스의 가중치(weight)  $W$ 는 자원 관리 장치에서 가지고 있는 GS와 CLS 예약을 그리고 현재 각 서비스 flow의 도착을 측정하여 동적으로 결정되며 각 서비스의 서비스율  $S$ 는 각 서비스의 도착율  $a$ 와  $W$ 의 최소값을 취하게 된다. 따라서 식 (11)에서와 같은 한 노드에서 전체 서비스를  $S_{tot}$ 는 전체 링크용량  $C_{tot}$ 를 초과하지 않게 된다. 위의 식을 [그림 5]와 같은 간단한 경우에 적용하였을 경우의 각 서비스에 할당된 대역폭 및 스케줄링의 결과는 다음과 같다.

$$GS : \min(4, 4) = 4$$

$$CLS : \min(4, 4) = 4$$

$$BE : \min(4, 2) = 2$$

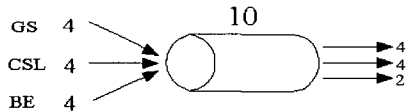


그림 5 WF2Q를 이용한 대역폭 할당

위의 예에서 GS와 CLS는 예약율과 같은 가중치를 얻기 때문에 서비스 flow의 도착율이 예약율과 같다면 요구되는 QoS를 만족시킬 수 있다. 이렇게 각 서비스의 가중치를 예약율에 따라 동적으로 결정하고 실제 각 서비스 flow의 도착율이 요구한 예약율을 초과하지 않는다면 요구한 QoS를 만족시킬 수 있다. 만약 실제 서비스 flow의 도착율이 예약율을 초과한다면 큐에 저장되어 도착율이 적을 때 처리되며, 측정된 다른 서비스 flow의 도착율이 적어 링크 용량이 남을 경우 측정된 서비스 도착율에 따라 각 서비스의 가중치를 동적으로 결정함으로써 예약율을 초과한 서비스 flow의 가중치를 높여 처리하게 된다.

### 3.3 제안방식의 노드 구조

[그림 6]은 위에서 제안된 수락제어와 QoS 스케줄링을 위한 edge 노드의 구조를 보여주고 있으며, 각 수행 블록의 기능은 다음과 같다.

- Classifier : 모든 노드들은 Intserv의 세 가지 서비

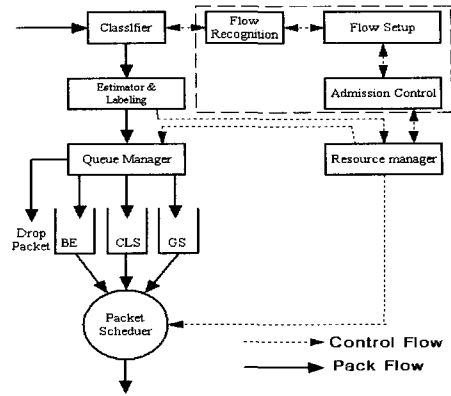


그림 6 제안 방식의 edge 노드 구조

스를 지원하기 위해서 필요한 기능을 할 수 있어야 한다. Classifier는 데이터 패킷이 어떠한 서비스 및 어느 flow에 속하는가를 확인 분류한다. IP 주소와 포트번호(TCP, UDP)는 패킷이 어느 flow 속하는지를 확인하는데 사용된다.

- Flow Recognition과 Flow Setup : GS 또는 CLS 서비스 요구 신호가 들어왔을 때 요구된 flow를 확인하고 flow 설정을 한 후, 요구된 QoS 정보를 수락제어 장치에 보내어 수락제어에 들어간다.
- Admission Control : 수락제어 장치는 이 QoS 요구 정보를 수락제어 신호를 요구된 QoS를 만족시킬 경우에 flow가 설정된다. 요구된 QoS를 만족시키지 못할 경우 수락은 이루어지지 않게 되고, 이 flow는 보다 낮은 QoS를 요구하거나 임의의 시간을 기다린 후에 다시 QoS를 요구하게 된다.
- Estimator & Labeling : 수락이 이루어진 flow들은 각 서비스 유형에 따라 분류되고 각 서비스 flow의 도착율을 측정하여 자원 관리 장치에 보내어 자원 관리에 이용하게 한다. 또한 패킷에 labeling을 수행한다.
- Queue Manager : Classifier에서 분류된 패킷을 각 서비스 유형에 따라 큐에 저장 관리한다.
- Packet Scheduler : 패킷 스케줄러는 WF2Q 방식을 이용해 최소한의 지연시간을 보장할 수 있도록 패킷 스케줄링을 수행한다.

[그림 7]은 제안 방식 수행을 위한 edge 노드의 구조를 보여주고 있다. Core 노드의 자원 관리 장치는 각 서비스에 할당되어 있는 자원을 관리하고 새로운 flow에 대한 예약을 요구하는 수락제어 신호가 들어왔을 때 노드의 자원이 이를 만족시킬 경우 자원을 할당하게

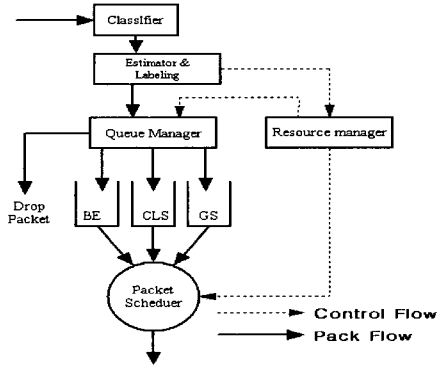


그림 7 제안 방식의 core 노드 구조

된다. 또한 core 노드의 자원 관리 장치는 각 서비스의 예약율과 전체 flow에 대한 정보만을 가지고 있으며 패킷 스케줄러는 자원 관리 장치에서 보내지는 각 서비스 flow의 예약율과 도착율을 통해서 각 서비스의 가중치를 얻고 패킷의 헤더에 표시된 flow의 상태 정보만을 가지고 스케줄링을 수행한다. 이 방법은 flow의 수가 증가하더라도 각 flow의 상태 정보와 각 flow 동작의 복잡성이 증가하지 않기 때문에 현재 제시된 Intserv 네트워크의 단점인 확장성을 해결할 수 있다.

#### 4. 성능 분석

SCORE 망을 기반으로 하는 제안 방식에 대한 성능을 시뮬레이션 통해 몇 가지 시나리오에 대하여 수행하였다. 성능분석은 Intserv 모델에서 각 서비스가 요구하는 QoS를 보장하기 위하여 필요한 각 flow의 대역폭 보장 여부 및 각 flow에 대한 패킷 지연과 패킷의 목적지 도착시간 간격, 즉 패킷 지연변이 관점에서 수행하였다. 시뮬레이션 도구로는 인터넷 프로토콜 시뮬레이션을 지원하는 NS-2(Network Simulator) 버전 2.1b6[11]을 이용하였다.

##### 4.1 시뮬레이션 모델

시뮬레이션은 제안된 방식과 FIFO 기반의 현재의 인터넷 방식에 대해서 동일한 환경변수를 이용하여 수행하였다. 시뮬레이션을 위한 망 구조는, 두 core 노드로 연결된 단순 구조(그림 8), 여러 core 노드가 mesh 형태로 연결된 복잡한 구조(그림 9) 및 계층적 구조(그림 10) 등 세 가지의 다양한 형태를 설정하여 제안 방식이 다양한 망 구조에서 효과적으로 동작되는지를 살펴보았다.

첫 번째 시뮬레이션 네트워크는 [그림 8]과 같이 두 core 노드로만 연결된 단순 구조이다. 각 노드는 n0부터

n10까지의 10 개의 송신 노드, n12부터 n21까지의 10개의 송신 노드, 그리고 n10과 n11의 core 노드로 구성되어 있다. 각 노드를 연결하고 있는 모든 링크는 10Mbps의 대역폭과 10ms의 지연 시간을 갖고 10개의 송신 노드는 1000bytes 길이의 패킷을 발생시킨다. 노드 n0, n1은 각각 GS와 CLS 서비스를 요구하는 트래픽을 발생시키고, 나머지 8개의 송신 노드는 BE 서비스를 요구하는 트래픽을 발생시키는 것으로 하였다. 10개의 송신 노드들이 발생시키는 전체 트래픽은 노드 n10과 노드 n11 사이의 링크 대역폭을 초과하도록 하여, 이러한 혼잡 링크 상에서 GS 및 CLS 서비스가 요구하는 대역폭을 보장받을 수 있는가와 GS 서비스 트래픽의 패킷 지연시간과 도착 시간 간격에 대한 실험을 수행하였다.

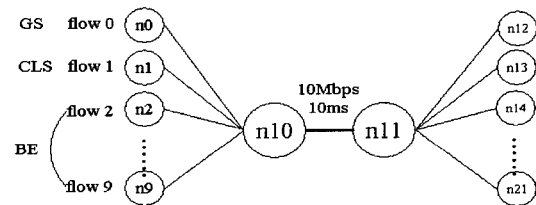


그림 8 서비스별 대역폭 할당 시뮬레이션 네트워크 구조

두 번째 시뮬레이션 네트워크는 [그림 9]와 같이 mesh 형태의 복잡한 구조이다. 노드 n0은 GS 서비스를 요구하는 트래픽을 발생시키며 나머지 노드들은 BE 서비스를 요구하는 트래픽을 발생시킨다. 각 노드를 연결하는 모든 링크는 10Mbps의 대역폭과 10ms의 지연 시간을 갖고 10개의 송신 노드는 1000bytes 길이의 패킷을 발생시킨다. 노드 n0에서 발생시킨 GS 서비스를 요구하는 트래픽에 대해서 혼잡 링크 수를 1부터 5까지 증가시키면서 대역폭을 보장받을 수 있는가를 실험하였다.

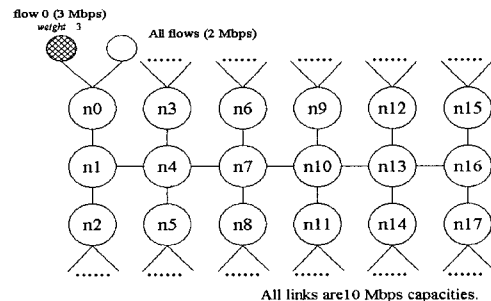


그림 9 혼잡 링크 상에서 대역폭 할당 시뮬레이션 네트워크 구조

세 번째 시뮬레이션 네트워크는 [그림 10]과 같이 계층 구조를 갖는다. 노드는 n0부터 n5까지의 송신 노드, n9부터 n14까지의 수신 노드, 그리고 n6, n7 및 n8의 core 노드들로 구성되어 있다. 각 노드를 연결하고 있는 링크는 10Mbps의 대역폭과 10ms의 지연 시간을 갖고 10개의 송신 노드는 1000bytes의 길이의 패킷을 발생시킨다. 노드 n0부터 n2까지는 노드 n6에 연결되고 노드 n3부터 n5까지는 노드 n7에 연결되고, 다시 노드 n6은 노드 n7에 연결되어 계층적인 네트워크 구조를 형성하고 있다. 이러한 계층적인 네트워크 구조에서 각 서비스에 따라 노드의 대역폭을 계층적으로 할당함으로써 각 서비스에 대한 QoS를 보장할 수 있는가를 성능 평가하였다.

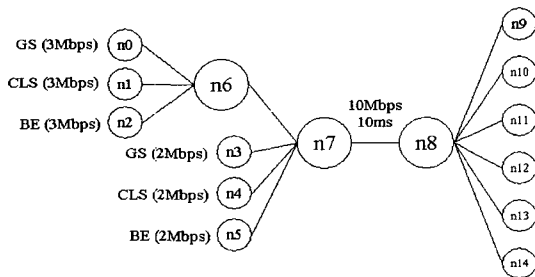


그림 10 계층적인 자원할당 시뮬레이션 네트워크 구조

4.2 시뮬레이션 결과

본 절에서 위에서 서술한 각 시뮬레이션 네트워크 구조에 따라 시뮬레이션을 환경 변수를 적용하여 수행한 결과를 서술한다.

4.2.1 서비스별 대역폭 분석

[그림 8]의 첫 번째 시뮬레이션 네트워크 구조에서 각 서비스가 할당받는 대역폭을 분석하기 위한 시뮬레이션 환경 변수는 [표 2]와 같다. 모든 서비스는 0.5초에 트래픽을 발생시켜 9초 동안 트래픽을 발생시킨다. 노드 n0는 보장형 서비스를 요구하는 3Mbps의 예약을 하고 3Mbps의 트래픽을 발생시키며, 노드 n1은 부하제어형 서비스를 요구하는 2Mbps의 예약을 하고 3Mbps

표 2 시뮬레이션 환경 변수 1

서비스 종류	트래픽 발생 시간	트래픽 발생율	예약율	패킷의 크기
보장형(GS)	9sec	3Mbps	3Mbps	1000bytes
부하제어형(CLS)	9sec	3Mbps	2Mbps	1000bytes
최선형(BE)	9sec	0.5, 0.6, 0.7, 0.75Mbps	없음	1000bytes

의 트래픽을 발생시킨다. 나머지 노드 n2에서 n9까지의 노드는 예약을 요구하지 않는 최선형 서비스 트래픽을 발생시킨다. 여기서 최선형 서비스 트래픽을 발생시키는 각 노드에서 트래픽 발생율을 0.5, 0.6, 0.7, 0.75Mbps로 증가시켜가며 보장형 서비스와 부하제어형 서비스가 예약한 대역폭을 보장받을 수 있는가를 실험하였다.

[그림 11]의 결과에서 n2에서 n9까지의 노드가 노드 n10과 n11의 링크 대역폭을 초과하지 않는 0.5Mbps의 트래픽을 발생시켰을 경우에 3Mbps의 트래픽을 발생시킨 보장형 서비스와 부하제어형 서비스는 모두 자신이 발생시킨 트래픽의 대역폭을 보장받고 있다. 하지만 최선형 서비스 트래픽이 각각 0.5Mbps 이상의 트래픽을 발생시켰을 경우에 전체 트래픽의 양은 노드 n10과 n11의 링크 대역폭 10Mbps를 초과한다. 따라서 3Mbps의 트래픽을 발생시킨 부하제어형 서비스는 최선형 서비스의 트래픽이 증가함에 따라서 2Mbps의 예약율까지 대역폭이 떨어진다. 하지만 최선형 서비스의 트래픽이 0.65Mbps 이상일 경우에도 보장형 서비스와 부하제어형 서비스는 모두 자신이 예약한 대역폭 3Mbps와 2Mbps를 보장받고 있는 것을 볼 수 있다. 따라서 제안된 방식의 네트워크에서 각 서비스가 요구하는 QoS를 보장함을 알 수 있다.

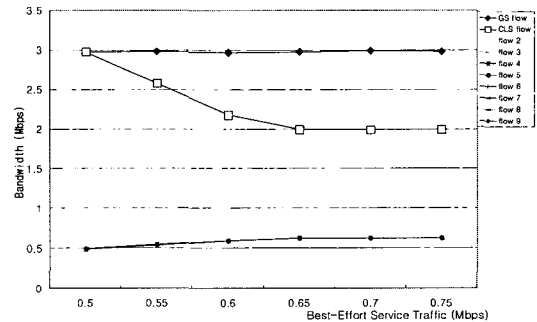


그림 11 BE 트래픽 증가에 따른 GS와 CLS의 대역폭 변화

두 번째 실험에서는 [그림 9]의 네트워크 구조에서 [표 3]과 같은 시뮬레이션 환경 변수를 설정하였다. 노드 n0는 보장형 서비스를 요구하는 3Mbps의 예약을 하고 3Mbps의 트래픽을 발생시키며, 노드 n1은 부하제어형 서비스를 요구하는 2Mbps의 예약을 하고 3Mbps의 트래픽을 발생시킨다. 나머지 노드 n2에서 n9까지의 노드는 예약을 요구하지 않는 최선형 서비스 트래픽을 3Mbps로 발생시킨다. 이러한 환경에서 제안 방식의



SCORE 네트워크 구조와 FIFO 기반의 현재의 인터넷 구조에서 각 서비스가 요구하는 대역폭이 어떻게 할당되는지에 대한 성능을 분석 비교하였다.

표 3 시뮬레이션 환경 변수 2

서비스 종류	트래픽 발생 시간	트래픽 발생율	예약율	패킷의 크기
보장형 (GS)	9sec	3Mbps	3Mbps	1000bytes
부하제어형 (CLS)	9sec	3Mbps	2Mbps	1000bytes
최선형(BE)	9sec	3Mbps	없음	1000bytes

[그림 12]와 [그림 13]은 각각 제안 방식의 SCORE 네트워크 구조와 FIFO 기반의 현재의 인터넷 구조에서 각 노드에 할당된 대역폭 모습을 나타내고 있다. 두 개의 그림에서 볼 수 있는 바와 같이 현재의 네트워크 구조에서는 보장형 서비스와 부하제어형 서비스가 요구하는 대역폭을 보장해 줄 수 없으나, 제안된 방식에서는 각 노드에서 요구하는 대역폭을 보장해 주는 것을 확인할 수 있다.

또한, 제안 방식이 실시간형 서비스에 적절함을 확인하기 위해, [그림 9]의 네트워크 구조에서 보장형 서비

스 패킷의 지연시간과 목적지 도착시간 간격, 즉 패킷 지연시간 변이의 성능을 알아보기 위한 실험을 수행하였다. [그림 14]는 노드 n0가 발생한 보장형 서비스 패킷의 지연시간의 결과를 보여주는데, 여기서 보장형 서비스는 예약한 대역폭을 보장받으며 패킷의 손실 없이 50 ms 이하의 end-to-end 패킷 지연시간을 갖음을 볼 수 있다. [그림 15]는 노드 n0가 발생한 보장형 서비스 패킷의 도착시간 간격을 측정한 결과로, GS 서비스를 요구하는 노드가 대역폭을 초과하지 않게 트래픽을 전송하면 제안된 방식의 네트워크에서는 대역폭이 보장되기 때문에 패킷의 손실이 없이 일정한 패킷 도착 시간 간격을 갖고 end-to-end 패킷 전송이 이루어짐을 확인할 수 있다. 따라서 제안된 방식의 네트워크는 최소한의 지연시간과 일정한 패킷 도착시간을 제공함으로써 실시간 서비스의 QoS가 보장됨을 확인할 수 있다.

4.2.2 혼잡 링크 수에 따른 대역폭 분석

[그림 9]의 시뮬레이션 네트워크 구조에서 [그림 8]의 혼잡링크 수를 1에서 5까지 증가시켜 갔을 때 보장형 서비스가 예약한 대역폭을 보장받을 수 있는가를 확인하는 실험을 [표 4]의 시뮬레이션 환경 변수를 설정하여 수행하였다. 보장형 서비스는 3Mbps의 대역폭을

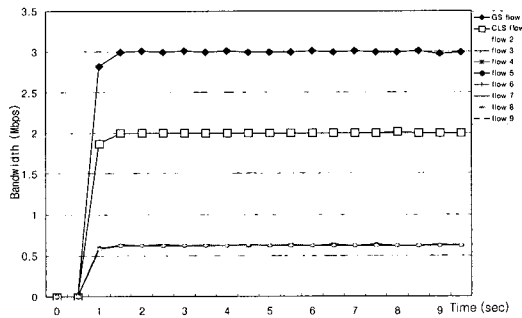


그림 12 제안 방식의 네트워크 대역폭 할당

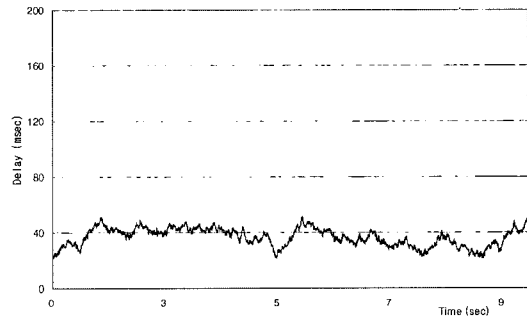


그림 14 GS 트래픽의 패킷 지연시간

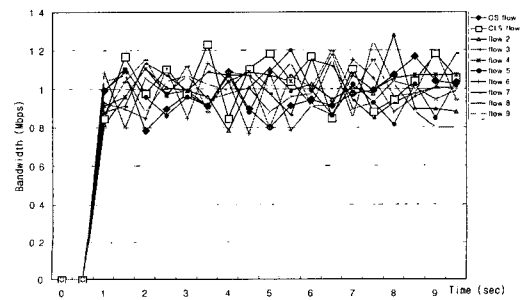


그림 13 FIFO를 이용한 현재의 네트워크 대역폭 할당

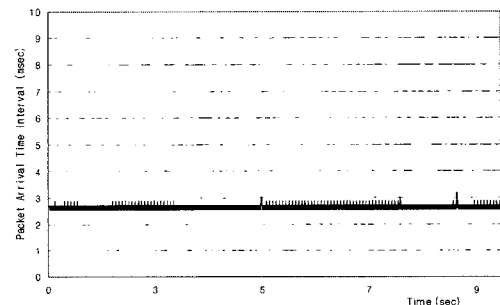


그림 15 GS 트래픽의 패킷 도착 시간 간격

예약하고 3Mbps의 트래픽을 발생시키며 최선형 서비스는 2Mbps의 트래픽을 발생시킨다. 따라서 전체 트래픽의 합은 10Mbps의 링크 대역을 초과하게 되고 이러한 링크 대역폭을 초과하는 혼잡 링크의 수를 1에서 5까지 증가시켜 가면서 보장형 서비스가 예약한 대역폭을 보장받을 수 있는가를 성능분석 하였다.

[그림 16]은 이 실험의 결과로 혼잡 링크 수가 증가하더라도 보장형 서비스가 예약한 대역폭이 보장됨을 보이고 있다. 이것은 제안된 방식의 네트워크 구조에서는 보장형 서비스가 예약한 대역폭은 많은 혼잡 링크가 존재할 경우에도 보장받을 수 있기 때문이다.

표 4 시뮬레이션 환경 변수 3

서비스 종류	보장형(GS), 최선형(BE)
트래픽 발생 시간	9sec
트래픽 발생율	3Mbps, 2Mbps
패킷의 크기	1000bytes
보장형 서비스의 예약율	3Mbps
링크 대역폭	10Mbps
혼잡 링크의 수	1, 2, 3, 4, 5

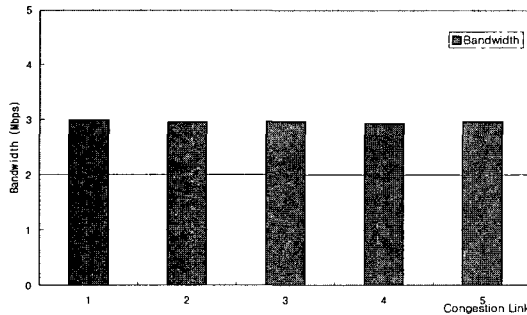


그림 16 혼잡 링크 수에 따른 GS 트래픽의 대역폭 할당

4.2.3 계층적인 네트워크 상에서 대역폭 분석

계층 구조를 갖는 망 구조에서도 제안 방식의 QoS 보장성을 알아보기 위해, [그림 10]의 계층적 망 구조에 대하여 실험을 수행하였으며, 각 서비스가 받는 대역폭을 분석하기 위한 시뮬레이션 환경 변수는 [표 5]와 같이 설정하였다. [그림 10]에서 보장형 서비스는 노드 n0와 n3에서 각각 3Mbps와 2Mbps의 트래픽을 발생시키며 각각 3Mbps와 2Mbps의 대역폭을 예약하고, 부하 제어형 서비스는 노드 n1과 n4에서 각각 3Mbps와 2Mbps의 트래픽을 발생시키며 각각 2Mbps의 대역폭을 예약하도록 하였다.

표 5 시뮬레이션 환경 변수 4

서비스 종류	트래픽 발생 시간	트래픽 발생율	예약율	패킷의 크기
보장형(GS)	9sec	3Mbps(flow 0), 2Mbps(flow 3)	3Mbps, 2Mbps	1000bytes
부하제어형(CLS)	9sec	3Mbps(flow 1), 2Mbps(flow 4)	2Mbps, 2Mbps	1000bytes
최선형(BE)	9sec	3Mbps (flow 2), 2Mbps(flow 5)	없음	1000bytes

[그림 17]의 실험 결과로부터, 보장형 서비스는 계층적인 자원 할당에 의하여 n0와 n3가 각각 예약한 대역폭 3Mbps와 2Mbps의 대역폭을 보장받고 있으며, 부하제어형 서비스의 경우는 자신이 예약한 대역폭을 초과한 n1의 3Mbps 트래픽은 예약한 대역폭 2Mbps만을 보장받고 예약한 대역폭을 초과하지 않은 n4의 경우는 2Mbps의 대역폭을 할당받음을 확인할 수 있다. 또한 최선형 서비스의 경우는 계층적인 대역폭 할당에 의하여 각각 1Mbps와 0.5Mbps의 대역폭을 할당받고 있다. 따라서 각 링크에 대해서 대역폭을 계층적으로 할당함으로써 각 서비스가 요구한 대역폭을 효율적으로 할당하고 있음을 보이고 있다.

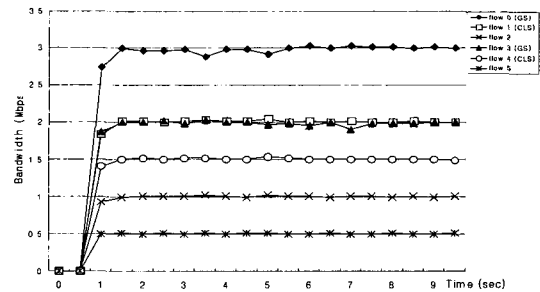


그림 17 계층적인 네트워크 상에서 대역폭 할당

5. 결론

본 논문에서는 IETF의 Intserv 서비스 모델의 확장성 문제를 해결하기 위해 제안된 단일 서비스 클래스의 SCORE 구조를 Intserv에서 제시한 세 가지 서비스 클래스에 적용시킬 수 있도록 확장한 내용을 다루었으며, 이를 위한 대역 할당 방법, WF2Q 스케줄링 방식 및 패킷 처리를 위한 edge 및 core 노드의 구조를 제시하였다. SCORE 구조를 기반으로 하는 본 제안 모델에서는 QoS를 지원하기 위한 각 flow에 대한 정보를 Intserv와 같이 각 노드가 관리 유지하는 것이 아니라, 패킷 헤더에 위치시키고 패킷이 망을 거쳐갈 때 이 정보가 노드의 상

항에 따라 갱신함으로써 core 노드의 구조를 간략화하여 망 확장성의 문제를 해결하도록 하였다.

또한 제안한 방식이 Intserv의 각 서비스가 요구하는 QoS를 보장하는지를 확인하기 위하여 ns-2 시뮬레이터를 이용하여 세 가지 네트워크 시나리오를 대상으로 성능분석 실험을 수행하였다. 이를 위한 성능변수로는 각 flow의 대역폭 할당, 패킷 지연 및 패킷의 목적지 도착 시간 간격을 사용하였다. 성능분석 결과, 제안 방식이 core 노드에서 각 flow의 상태 정보를 유지하는 stateful 네트워크와 같이 충분히 서비스 품질을 보장할 수 있으며, 이를 통해 core 노드 구조를 단순화시키는 제안 방식이 Intserv 모델의 단점인 확장성을 문제를 해결할 수 있는 좋은 방안임을 확인하였다.

### 참고 문헌

- [1] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview," Internet RFC 1633, June 1994.
- [2] S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service," Internet RFC 2212, September 1997.
- [3] J. Wroclawski, "Specification of the Controlled-Load Network Element Service," Internet RFC 2211, September 1997.
- [4] R. Braden, Ed., L. Zhan, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification," RFC 2205, September 1997.
- [5] Ion Stoica, Hui Zhang, "Providing Guaranteed Services Without Per Flow Management," ACM SIGCOMM'99, Boston, MA, September 1999.
- [6] Ion Stoica, Scott Shenker, Hui Zhang, "Core-Stateless Fair Queueing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks," SIGCOMM'98.
- [7] S. Blake, et. al., "An Architecture for Differentiated Services," RFC 2475, December 1998.
- [8] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queing Algorithm," ACM Computer Communication Review, vol. 25, pp.174-187, Jan. 1995.
- [9] J.C.R. Bennett and H. Zhang, "Why WFQ Is Not Good Enough for Integrated Services Networks," Proceedings of NOSSDAV'96, Apr, 1996.
- [10] J.C.R. Bennett and H. Zhang, "WF2Q: Worst-case fair weighted fair queueing," In Proceedings of IEEE INFOCOM'96, San Francisco, CA, March 1996.
- [11] <http://www.isi.edu/nsnam/ns/>



민 안 기

1999년 2월 연세대학교 전산학과(학사).  
2001년 2월 연세대학교 전산학과(석사).  
2001년 3월 ~ 현재 (주) 씨앤드씨 엔지니어링 연구원. 관심분야는 Network 트래픽 및 QoS 제어, Network Hardware 및 Embedded System 설계



석 정 봉

1979년 2월 연세대학교 전자공학과(학사). 1981년 2월 한국과학기술원 전기 및 전자공학과(석사). 1989년 9월 Univ. of Massachusetts 전기 및 컴퓨터공학과(박사). 1981년 3월 ~ 1993년 2월 한국전자통신연구원 책임연구원. 1993년 3월 ~ 현재 연세대학교 전산학과 교수. 관심분야는 컴퓨터 통신망 모델링 및 최적화, Internet/ATM/무선통신망의 트래픽 및 QoS 제어