

QoS 라우팅과 함께 효율적인 최선트래픽 라우팅을 지원하는 서버 기반 라우팅 기법

최 미 라[†] · 김 성 하^{††} · 이 미 정^{†††}

요 약

QoS 라우팅은 네트워크 활용률 및 사용자의 서비스 수준을 향상시키는 효과가 있으나 기존의 최선 라우팅보다 훨씬 복잡하고 빈번한 경로 계산 및 동적 네트워크 상태 정보 교환을 위한 프로토콜 오버헤드를 발생시킨다. 또한 네트워크에 QoS 라우팅이 도입된 경우 상대적으로 우선 순위가 낮은 최선 트래픽에 대한 네트워크 성능을 향상시킬 수 있는 방안에 대해서는 거의 연구된 바가 없다. 이에 본 논문에서는 QoS 라우팅을 위한 동적 네트워크 상태 정보 교환 오버헤드 없이 QoS 라우팅을 수행하고 QoS 라우팅 성능에 전혀 영향을 주지 않으면서 우선 순위가 낮은 최선 트래픽의 성능 또한 향상시킬 수 있는 서버 기반 라우팅 메커니즘을 제안한다. 시뮬레이션을 통해 제안한 라우팅 방안의 성능을 평가한 결과, 제안하는 라우팅 방안이 기존의 QoS 라우팅에 비해 더 적은 QoS 라우팅 오버헤드를 발생시키면서도 QoS 트래픽에 대한 라우팅 성능을 향상시킬 뿐 아니라, 최선 트래픽에 대한 라우팅 성능도 향상시킬 수 있었다. 특히, 최선 트래픽에 대한 QoS 트래픽의 영향이 큰 경우 즉, 네트워크의 전체 트래픽에 대한 QoS 트래픽의 비중이 높은 경우에 더욱 효과적으로 최선 트래픽의 성능을 향상시킬 수 있었다.

A Server Based Routing Mechanism Providing QoS Routing with Efficient Support of Best Effort Traffic

Mira Choe[†] · Sungha Kim^{††} · Meejeong Lee^{†††}

ABSTRACT

QoS routing can improve network performance while providing support for QoS guarantees. These benefits, however, comes with additional routing costs such as more complex and frequent route computation and the protocol overheads to exchange dynamic network state information. Moreover, little has been done to minimize the impact of the QoS traffic to the best effort traffic or to enhance the routing performance of the best effort traffic when QoS routing is deployed. In this paper, it is proposed that a server based routing mechanism, which supports the QoS routing without incurring the QoS routing protocol overhead for the network state update exchanges and enhances the performance of the best effort traffic without affecting the performance of QoS routing. Simulation results show that the proposed scheme enhances the routing performance for the QoS traffic while reducing the routing protocol overhead. The routing performance of the best traffic is also improved with virtually no impact to the routing performance of the QoS traffic. The proposed scheme is shown to be especially effective when the ratio of QoS traffic is high, that is, when the impact of the QoS traffic to the performance of best effort traffic is significant.

키워드 : 서비스 품질(Quality of Service ; QoS), 서비스 품질 지원 라우팅(QoS Routing), 최선 서비스 라우팅(Best Effort Routing), 다중 서비스 라우팅(Multi-Service Routing)

1. 서 론

패킷 스위칭 네트워크의 용량이 커짐에 따라, 비디오 컨퍼런스나 인터넷 전화, 원격 학습 등의 새로운 응용들을 지원할 수 있게 되었다. 그런데 이들 응용은 기존의 데이터 트래픽과는 달리 QoS(Quality of Service) 요구를 네트워크에 부과한다. 이에, 최근 동적 매트릭에 근거한 QoS 라우팅

에 관한 활발한 연구가 이루어져 왔다. QoS 라우팅이란, 사용자 플로우가 요구하는 QoS를 지원할 수 있는 '적정 경로'를 동적으로 결정하는 라우팅으로서, 네트워크 활용률 및 사용자 서비스 수준을 향상시킬 수 있다. 그러나 이와 같은 QoS 라우팅을 위해서는 복잡하고 잦은 경로 계산과 경로 계산에 필요한 네트워크 자원의 상태 정보를 갱신하기 위한 라우팅 프로토콜 오버헤드 등이 요구된다.

또한, 실제로 네트워크 상에는 QoS 트래픽뿐 아니라 전통적인 최선 트래픽이 공존하게 되는데, 이들 두 트래픽 클래스간에 효율적인 자원 공유를 지원할 수 있는 방안이 필요하다. 다중의 트래픽 클래스 지원을 위해 트래픽 클래스

* 본 연구는 한국과학재단 목적기초연구(R04-2000-000-00078) 지원으로 수행되었음.

† 준 회 원 : 이화여자대학교 컴퓨터학과

†† 정 회 원 : 이화여자대학교 컴퓨터학과

††† 정 회 원 : 이화여자대학교 컴퓨터학과 교수

논문접수 : 2002년 8월 22일, 심사완료 : 2002년 10월 19일

별로 네트워크 자원을 할당할 수도 있으나, 트래픽 클래스 간에 네트워크 자원을 공유하도록 하고 라우팅에서 클래스 간 효율적인 네트워크 자원 공유를 고려한 경로 결정이 이루어진다면 네트워크 자원 활용을 향상시킬 수 있다. 이와 같은 경우, 서로 다른 서비스 클래스에 속하는 트래픽을 우선 순위에 근거하여 라우팅하고 포워딩하면서도, 우선 순위가 낮은 트래픽에 가용한 네트워크 자원의 고갈을 막고 서비스 클래스들을 모두 효율적으로 지원할 수 있는 방안이 필요하다.

최근 QoS 라우팅의 추가적인 비용을 감소시키고자 하는 측면에서 연구가 활발히 진행되어 경로 계산 오버헤드를 감소시키기 위한 사전 경로 계산(path pre-computation) 및 경로 캐싱(path caching) 등의 방안이 제안되었고[1-7], 네트워크 상태 정보 갱신에 관련된 라우팅 프로토콜 오버헤드를 최소화하기 위한 다양한 네트워크 상태정보 갱신 유발 정책들이 제안되었다[8,9]. 또한, QoS 라우팅에 관한 대부분의 작업들을 모두 서버가 수행하도록 함으로써, 다른 라우터들이 복잡한 QoS 라우팅의 오버헤드를 부담하지 않아도 되도록 하는 서버 기반 QoS 라우팅 방식도 제안되었다[10]. 이들 기존의 QoS 라우팅 스킴에서는 공통적으로 QoS 경로 계산에 필요한 네트워크 상태 정보를 얻기 위하여 라우터들 간에 정기적 혹은 특별한 동적 변화가 있을 때마다 네트워크 상태 정보를 서로 교환한다[1-10]. 그런데, 이런 QoS 라우팅 패러다임에서는 라우팅 성능의 향상을 위해 빈번한 네트워크 상태 정보 교환이 이루어져야 한다. 더 나아가, 네트워크 상태 정보가 QoS 매트릭 상태에 변화가 있을 때마다 갱신되지 않는다면 근본적으로 부정확한 네트워크 상태 정보를 바탕으로 라우팅 결정을 내리게 된다.

또한, QoS 라우팅이 적용되는 경우 최선 트래픽의 성능을 향상시키기 위한 방안들이 제안된 바 있는데[11,12], 이들 방안은 QoS 트래픽으로 하여금 최선 트래픽으로 인해 부하가 높은 지역을 우회하도록 하여 QoS 트래픽으로 인한 최선 트래픽의 자원 고갈을 막고, 성능 저하를 감소시킨다. 그러나 이와 같은 방법은 QoS 트래픽이 최선 트래픽을 우회하여 라우팅되도록 하기 때문에 QoS 트래픽을 우선적으로 지원한다는 정책에는 적합하지 않다. 이들 연구의 수치적 결과에서도 이와 같은 정책을 씀으로 인해 QoS 트래픽 플로우의 블로킹율이 증가함을 볼 수 있다[11,12].

이에, 본 논문에서는 동적인 네트워크 상태 정보 교환 오버헤드 없이 QoS 라우팅을 수행하고 QoS 트래픽에 전혀 영향을 주지 않으면서 최선 트래픽의 성능을 향상시키는 서버 기반의 라우팅 방안을 제안한다. 제안하는 라우팅 방안에서는 각 네트워크 엔티티에서 QoS 트래픽과 최선 트래픽을 위한 자원이 사전에 정적으로 프로비전되는 것이 아니라 QoS 플로우가 발생하여 그 경로가 결정됨에 따라 QoS 플로우를 위한 자원이 예약되고 QoS 플로우는 예약된 자원을

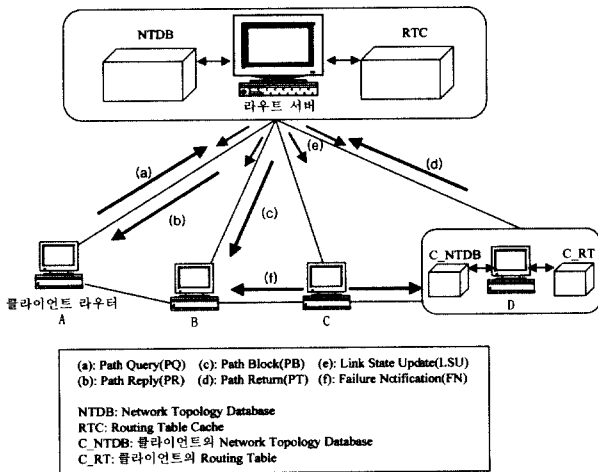
이용하여 서비스되며 최선 트래픽은 나머지 네트워크 자원을 이용하여 서비스된다고 가정한다. 제안하는 서버 기반 라우팅을 위해, [10]에서와 마찬가지로 라우팅 도메인 내에는 하나의 라우트 서버를 두고 이 라우트 서버에서 라우팅 도메인 안의 모든 QoS 라우팅을 담당하도록 한다. 그러나, 라우트 서버는 [10]에서와 같이 클라이언트 라우터들과의 네트워크 상태 정보 교환을 통해 네트워크 상태 정보를 수집하지 않고, 클라이언트들의 요청에 따라 QoS 플로우에 대하여 QoS 경로를 할당하거나 QoS 경로를 다시 반환 받으면서 이에 따른 네트워크의 가용 자원 양의 변화를 자신이 유지하고 있는 네트워크 상태 정보 데이터베이스에 반영함으로써 정확한 네트워크 상태 정보를 유지한다. 따라서, 제안하는 서버 기반 QoS 라우팅에서는 기존의 QoS 라우팅에서와 같은 네트워크 상태 정보 교환을 위해 플러딩되는 프로토콜 오버헤드가 없으며, 정확한 네트워크 상태 정보를 이용해 경로를 계산하므로 라우팅 성능을 향상시킨다.

한편 최선 트래픽에 대해서는 라우트 서버의 관여 없이 각 진입 라우터에서 명시적인 소스 라우팅을 수행한다. 단, 라우트 서버의 QoS 트래픽 경로 할당에 따라 결정되는 QoS 트래픽의 부하 분포를 반영해서 진입 라우터들이 최선 트래픽을 라우트하도록 하기 위해 정기적으로 QoS 트래픽 부하 분포에 대하여 라우트 서버가 클라이언트 라우터들에게 방송되도록 한다. 즉, 제안하는 라우팅 방안에서는 QoS 트래픽으로 인해 부하가 높은 지역을 최선 트래픽이 우회하게 되며, 결과적으로 QoS 트래픽의 라우팅 성능에 영향을 주지 않으면서 최선 트래픽의 성능을 향상시킬 수 있다. 즉, 분산 QoS 라우팅이 이루어지는 상황에서 최선 트래픽의 성능을 향상시키기 위해 제안된 기존의 방식들은 QoS 트래픽이 최선 트래픽을 우회하여 경로를 선택하도록 하는 반면[11,12] 제안하는 방안은 최선 트래픽이 QoS 트래픽을 우회하도록 하는 것이다. 또한, 최선 트래픽만 존재하는 경우 부하 균등화 등을 위해 네트워크의 동적인 정보를 사용하는 기존의 최선 트래픽 라우팅 방식과 비교해 보면, 이들 방식에서는 대부분 모든 라우팅 노드들이 네트워크의 동적 상태 정보를 알리는 제어 메시지를 발생하도록 하는데 반하여[13], 제안하는 방식에서는 라우트 서버만이 이를 발생시킬 뿐 아니라 이와 같은 프로토콜의 효율성을 높이기 위한 방안 즉, 제어 메시지 발생량은 줄이면서 라우팅 성능 퇴보는 가능한 한 방지할 수 있는 방법도 두 가지로 제안하였다.

본 논문의 구성은 다음과 같다. 1장 서론에 이어 2장에서는 제안하는 다중 서비스 클래스 지원 라우팅 방안에 대하여 자세히 설명한다. 3장에서는 제안하는 방안의 성능 평가 및 기존 연구와의 비교를 위한 시뮬레이션 모델을 설명하고, 4장에서는 시뮬레이션 결과를 분석한다. 마지막으로 5장에서는 본 논문의 결론을 기술한다.

2. 다중 서비스 클래스 지원을 위한 라우팅 방안

(그림 1)은 제안하는 다중 서비스 클래스 지원 라우팅을 위한 네트워크 구성 요소 및 이들 간의 통신을 도식화한 것이다. 제안하는 다중 서비스 클래스 지원 라우팅을 수행하는 네트워크는 (그림 1)에서와 같이 라우트 서버와 클라이언트 두 가지 요소로 구성된다. 라우팅 도메인에는 라우트 서버 역할을 담당하는 디바이스가 하나 존재하고 다른 모든 라우터들은 클라이언트가 된다. 네트워크 활용률 향상시키기 위해 QoS 트래픽과 최선 트래픽은 동적으로 네트워크 자원을 공유하고 사전에 정적으로 프로비전되지 않는다고 가정한다.



(그림 1) 제안하는 서버 기반 라우팅 방안을 위한 네트워크 구성 요소 및 이들 간의 통신

라우트 서버는 네트워크 토폴로지 정보 및 모든 링크들의 링크 QoS 상태 정보를 유지하는 NTDB(Network Topology Data Base)와 QoS 경로 정보를 일시적으로 저장하여 재 사용할 수 있도록 하는 RTC(Routing Table Cache)를 이용하여 클라이언트의 QoS 경로 질의에 대해 응답한다. 이에 관련된 각 메시지의 사용 및 처리는 다음과 같다.

- PQ(Path Query) : 클라이언트 라우터에 새로운 QoS 플로우의 전송 요구가 발생하면 클라이언트는 라우트 서버에게 해당 QoS 요구사항을 실은 PQ 메시지를 보낸다(그림 1)(a). 이 PQ 메시지를 받은 라우트 서버는 NTDB의 현재 네트워크 토폴로지 상태 정보를 기반으로 QoS 요구사항을 만족하는 경로를 새로 계산하거나, RTC를 검색하여 적정 경로 정보를 선택하여 클라이언트에게 PR(Path Reply) 메시지로 응답한다(그림 1)(b).
- PR(Path reply) : PR 메시지는 클라이언트가 PQ를 이용해 질의한 목적지에 대하여 명시적인 소스 라우팅 정보를 알려준다. 라우트 서버는 클라이언트에게 PR 메

시지를 발송하면서 해당 경로에 대한 자원 할당 사실을 NTDB에 반영하기 위해 NTDB를 갱신한다.

- PB(Path Block) : PQ 메시지에 대하여 만약 적정 경로가 존재하지 않는다면 라우트 서버는 PB 메시지를 클라이언트에게 보낸다(그림 1)(c).
- PT(Path Return) : QoS 플로우의 전송이 끝나면 클라이언트는 이를 라우트 서버에게 보고하기 위해 PT 메시지를 보낸다(그림 1)(d). PT는 라우트 서버가 할당했던 경로의 자원이 다시 가용해졌음을 의미하므로 라우트 서버는 PR 메시지를 보낼 때와 마찬가지로 반환된 경로 상의 가용 자원 정보 변화를 반영하기 위해 NTDB를 갱신한다.

위의 메시지들을 교환하는 클라이언트/서버 교류는 신뢰성 있게 이루어져야 하는데, 이를 위해 COPS(Common Open Policy Service) 프로토콜을 이용해 이들 메시지를 교환할 수 있다. COPS 프로토콜은 정책 기반 네트워크 환경에서 클라이언트와 서버의 신뢰성 있는 메시지 교환을 위해 정의되었다[14].

한편, 최선 트래픽 라우팅을 위하여 각 클라이언트 라우터들은 네트워크 토폴로지 및 QoS 트래픽의 분포에 의한 네트워크 상태 정보를 저장하는 C_NTDDB와 이를 기반으로 구축한 라우팅 테이블인 C_RT를 유지한다. 클라이언트 라우터들은 라우트 서버로부터 QoS 트래픽 분포로 인한 네트워크 상태 정보를 가르쳐 주는 LSU(Link State Update) 메시지(그림 1)(e)를 받아 이에 따라 C_NTDDB를 갱신·유지하고, C_NTDDB로부터 각 목적지로의 경로를 계산해 C_RT에 저장한다. 임의의 최선 트래픽 플로우에 대한 진입 클라이언트 라우터는 C_RT에서 해당 목적지로의 경로를 발견하고 이 경로로 패킷을 전송한다. 또한, 모든 라우터들은 자신에게 인접한 링크나 노드에서 실패가 발생한 경우 이를 기존의 OSPF와 같은 방식으로 네트워크 전체에 FN(Failure Notification) 메시지를 플러딩함으로써 라우트 서버 및 다른 모든 클라이언트 라우터들이 각각 NTDB 및 C_NTDDB와 RTC 및 C_RT를 갱신하도록 한다(그림 1)(f).

이와 같은 라우팅 방안에서 가장 문제가 되는 프로토콜 오버헤드는 라우트 서버와 클라이언트 라우터간에 QoS 경로 질의와 이에 대한 응답을 위해 교환하는 프로토콜 제어 메시지 오버헤드이다. 라우트 서버가 정기적으로 LSU를 플러딩하기 위한 프로토콜 오버헤드도 있지만 이것은 OSPF에 근거한 QoS 라우팅에서 모든 라우터가 상태 정보를 플러딩하는 것에 비할때 매우 적은 양의 오버헤드에 불과하다. 이에 본 논문에서는 라우트 서버와 클라이언트 간에 QoS 라우팅을 위해 교환하는 프로토콜 메시지 오버헤드가 어느 정도인지를 보기 위하여 시뮬레이션을 통해 이를 기존의 분산 QoS 라우팅에서의 네트워크 상태 정보 교환을 위해 플

러딩되는 프로토콜 오버헤드와 비교하였다. 이와 같은 프로토콜 오버헤드와 함께 고려되어야 할 오버헤드는 병목 지점이 될수 있는 라우트 서버에서의 프로세싱 및 메모리 오버헤드이다. 이에 2.1에서는 라우트 서버에서의 QoS 경로 계산 오버헤드를 감소시키기 위한 방안으로서 경로 캐싱 방법을 제안한다. 그리고, 2.2절에서는 최선 트래픽을 위한 라우팅 방안에 대하여 자세히 설명한다.

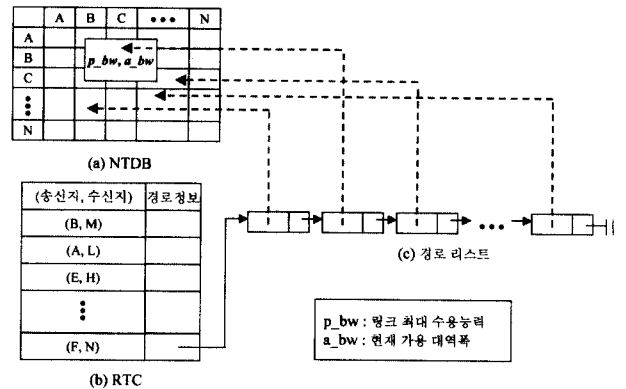
2.1 서버 기반 QoS 라우팅

제안하는 서버 기반 QoS 라우팅에서는 “QoS를 위한 OSPF 확장 프로토콜”과 마찬가지로 QoS 요구가 최소 대역폭 요구로 주어진다고 가정하고, 링크 상태(link state) 라우팅에 기반 하여 적정 경로를 계산한다 [2]. 따라서 라우트 서버가 파악해야 하는 QoS 매트릭은 각 링크의 가용 대역폭이다. 경로 계산 알고리즘으로는 변형된 Dijkstra 최단 경로 알고리즘 혹은 QoS를 위해 확장된 Bellman Ford 알고리즘을 사용한다[2]. 변형된 Dijkstra 알고리즘이란 QoS 경로를 계산하는 시점에 링크의 가용 대역폭이 클라이언트가 요구한 대역폭보다 작게 남아 있는 링크들을 모두 제거한 후 남은 네트워크 토폴로지 상에 Dijkstra 알고리즘을 적용하여 widest-shortest 경로를 계산하는 것을 말한다. Bellman Ford 알고리즘은 모든 가능한 홉 수에 대하여 홉 수별로 최대 대역폭 경로를 계산하되 최소한 더 짧은 경로에서 제공할 수 있는 대역폭보다 더 큰 대역폭을 제공할 수 있는 경로만을 계산한다.

네트워크 상에서 일어나는 통신은 송·수신지 쌍에 있어서 상당한 지역성이 있다. 만약 제안하는 서버 기반 라우팅에서 라우트 서버가 사전 경로 계산 정책에 기반하여 라우팅 테이블을 미리 계산하여 유지한다면, m 을 라우팅 도메인내의 모든 노드의 수라고 가정했을 때 m^2 개의 송·수신지 쌍에 대한 QoS 경로를 모두 라우팅 테이블에 유지해야 한다. 그러나 통신 형태의 지역성으로 인해 이 중 상당수의 라우팅 테이블 엔트리들은 거의 혹은 전혀 사용되지 않게 될 것이다. 반대로 이러한 통신 형태의 지역성에 의해 적은 수의 캐시 엔트리만으로도 효과적으로 라우트 서버의 경로 계산 오버헤드를 줄일 수 있다. 따라서 본 논문에서는 라우트 서버의 오버헤드를 줄이기 위해서 경로 캐싱 방식을 채택하였다.

RTC에는 한번 계산된 QoS 경로 정보를 임시로 저장하는데 기존 QoS 라우트 캐싱 스킴에서 경로 정보와 함께 경로의 QoS 정보를 저장하는 것과는 달리 제안하는 라우팅 방안에서는 RTC에 경로를 구성하고 있는 링크들의 리스트만을 유지한다. 즉, 병목 대역폭과 같은 해당 경로의 QoS 상태 정보는 따로 저장하지 않고, 클라이언트로부터 경로 요청이 들어와 RTC에서 적정 경로를 선택할 때마다 현재 네트워크 상태에 대한 정확한 정보를 유지하고 있는 NTDB

를 검색함으로써 파악한다. 임의의 클라이언트로부터 특정 수신지로의 QoS 경로를 요청 받은 라우트 서버는 우선 현재 RTC에 해당 요구에 대한 QoS 경로 정보가 이미 존재하는지 판단하기 위해 RTC를 검색한다. RTC에 해당 경로 정보가 존재한다면 해당 경로를 구성하고 있는 모든 링크들에 대해 NTDB에서의 현재 가용 대역폭을 검색하여 그 경로의 현재 병목 대역폭을 확인한다. 라우트 서버의 RTC 및 NTDB 검색 작업을 효율적으로 하기 위해 각 자료 구조는 (그림 2)에서와 같은 형태를 갖는다. RTC의 각 엔트리에서 유지하고 있는 QoS 경로 정보는 해당 QoS 경로를 구성하고 있는 각 링크들의 NTDB에서의 엔트리를 가리키는 포인터들의 연결 리스트로 되어 있다. 라우트 서버는 이 포인터를 따라감으로써 해당 QoS 경로를 구성하고 있는 각 링크에 대해 NTDB에서 현재 가용 대역폭을 확인할 수 있다.



(그림 2) 라우트 서버가 유지하는 NTDB와 RTC 구조

RTC의 크기는 K 와 n 두 가지의 파라미터로 제한된다. K 는 RTC에서 유지할 수 있는 송·수신지 쌍의 수이고, n 은 임의의 송·수신지 쌍에 해당하는 엔트리에서 유지할 수 있는 서로 다른 경로 정보의 개수를 말한다. 앞으로 하나의 송·수신지 쌍에 대해 n 개의 경로를 저장하는 스킴을 n -경로캐싱이라 부르기로 한다. 클라이언트로부터 경로 요청이 들어오면 RTC에서 해당 송·수신지 쌍 엔트리에 저장되어 있는 n 개의 경로들 중 최단 경로부터 우선적으로 선택하여 적정성을 판단한다. 만약 해당 송·수신지 쌍에 대한 엔트리의 n 개의 경로들 중 적정한 경로가 하나도 존재하지 않는다면 라우트 서버는 현재의 NTDB 정보를 기반으로 새로운 QoS 경로를 계산한다. 변형된 Dijkstra 알고리즘을 사용하는 경우는 현재 요구를 만족시키는 단 하나의 경로가 계산되는데, RTC의 해당 송·수신지 엔트리에 현재 저장되어 있는 경로 정보 수가 n 보다 작다면 새로 계산된 경로 정보는 단순히 그 RTC 엔트리에 추가된다. 그러나 해당 송·수신지에 해당하는 RTC 엔트리에 현재 n 개의 경로 정보가 이미 저장되어 있다면 이중 하나를 선택하여 새로이 계산된 경로 정보와 교체해야 한다. 이와 관련된 정책

을 경로 수준의 교체 정책이라 한다. 제안하는 스킴에서는 잔여 병목 대역폭의 크기가 가장 적은 경로들 중에 가장 긴 경로를 희생자로 선택한다. QoS 라우팅을 위한 Bellman-Ford 알고리즘을 이용해 경로를 계산하는 경우는 n 개의 경로를 모두 계산하게 되므로, 해당 송·수신지 엔트리의 경로 정보를 모두 대체한다.

RTC에 클라이언트가 요구한 송·수신지 쌍에 대한 엔트리가 존재하지 않는 경우 NTDB의 네트워크 토폴로지 정보를 기반으로 이 새로운 송·수신지 쌍에 대한 QoS 경로를 계산하여 이를 RTC에 추가해야 한다. 만약 RTC에 이미 K 개의 송·수신지 쌍에 대한 엔트리가 차있는 경우에는 이중 한 엔트리를 선택하여 새로운 송·수신지 쌍과 교체해야 한다. 이와 관련된 정책을 송·수신지 수준의 교체 정책이라 한다. 라우팅 도메인 내의 통신 형태에 있어서 통계적인 지역성이 존재한다면 캐싱 스킴은 이런 지역성을 송·수신지 수준의 교체 정책에 반영할 수 있어야 한다. 이를 위해 제안하는 스킴에서는 RTC의 각 송·수신지 쌍 엔트리마다 참조된 횟수를 나타내는 카운터를 유지하고 이 카운터 값이 가장 적은 엔트리를 희생자로 선택한다. 또한 정기적으로 카운터 값을 오른쪽으로 쉬프트함으로써 한동안 빈번하게 참조되었으나 더 이상 QoS 플로우 발생이 없는 송·수신지 쌍이 계속해서 RTC 엔트리를 차지하는 것을 방지하였다.

2.2 최선 라우팅 방안

최선 트래픽을 위해서는 각 진입 경계 라우터(ingress edge router)에서 목적지까지의 경로를 결정하고 명시하는 소스 라우팅을 사용한다. 이것은 MPLS에 부합하는 개념으로 MPLS 네트워크에 도입될 경우 최선 트래픽 클래스에 속하는 플로우를 위한 LSP 설정을 위해 제안하는 라우팅 방안을 사용하고, 각 패킷에는 해당 LSP의 레이블을 붙여서 전송하면 된다.

기존의 최선 트래픽을 위한 라우팅에서는 네트워크 상태 변화에 관계없이 항상 최단 경로로 트래픽을 전송하는데, 이와 같은 최단 경로 라우팅은 최단 경로 상에 트래픽 부하가 높은 경우, 다른 지역에는 가용 자원이 있음에도 불구하고 긴 전송 지연과 손실을 겪게 된다. 특히, QoS 트래픽과 최선 트래픽이 네트워크 자원을 공유하는 경우, 우선 순위가 낮은 최선 트래픽의 성능 저하가 매우 클 수 있다.

제안하는 라우팅 방안에서는 라우트 서버가 항상 정확하게 네트워크 상에 QoS 트래픽이 어떻게 분포되어 있는지를 파악하고 있다는 점을 이용하여, 라우트 서버가 이 정보를 클라이언트 라우터들에게 알려줌으로써 최선 트래픽 라우팅 결정에 QoS 트래픽의 부하 분포 정보를 반영할 수 있도록 하였다. 클라이언트 라우터는 초기화시 항상 라우트 서버로부터 네트워크 전체에 대한 가용 자원 상태 정보를 얻는다.

또한 라우트 서버는 정기적으로 클라이언트들에게 QoS 트래픽 분포에 대한 네트워크 상태 정보를 LSU에 실어 브로드캐스트하는데, 이때 라우트 서버가 LSU에 포함할 대상 링크를 선정하는 방식 및 브로드캐스트시 포함되는 정보, 그리고 클라이언트 라우터들이 이를 유지하는 형태 등에 따라 크게 다음과 같은 두 가지 방안을 제안한다.

첫 번째 방안에서는 QoS 트래픽에 의한 링크 활용률(U_i)이 임의의 임계치 이상으로 높은 링크들을 네트워크 전체에 알리는데 이를 '절대 임계치 갱신'이라 한다. 이 경우 라우트 서버는 LSU 메시지에 해당 링크들의 아이디만을 표시하여 플러딩한다. 두 번째 방안에서는 라우트 서버가 지난번에 자신이 선정한 링크의 가용 대역폭과 현재 가용 대역폭의 차이가 임계치보다 큰 링크들에 대해서만 새로운 가용 대역폭 정보를 플러딩하는데, 이를 '상대 임계치 갱신'이라고 한다[9]. 이 경우 라우트 서버는 LSU 메시지에 대상이 되는 모든 링크들에 대하여 아이디뿐만 아니라 현재 가용 대역폭 정보를 표시한다.

클라이언트 라우터들은 라우트 서버가 보낸 LSU 메시지를 받으면 C_NTDB를 갱신하고 이를 기반으로 다시 C_RT를 계산한다. 두 스킴은 LSU 메시지에 기입하는 정보가 다르기 때문에 이로 인해 각각 다른 경로 계산 알고리즘을 사용한다. 절대 임계치 갱신 스킴에서는 각 클라이언트 라우터들이 최선 트래픽 전송을 위한 경로 계산시 LSU에서 표시된 활용률이 높은 링크들을 먼저 제거하고 Dijkstra 최단 경로 알고리즘을 적용하여 최단 경로를 계산한다. 라우트 서버가 알려준 활용률이 높은 링크들을 경로 계산에서 제외시킴으로써, 클라이언트 라우터들은 QoS 트래픽으로 인해 자원이 고갈된 지역을 효과적으로 우회하여 최선 트래픽을 라우팅할 수 있다. 이 라우팅 스킴은 단순하고 오버헤드가 낮은 장점을 갖고 있다. 그러나 모든 클라이언트 라우터들이 같은 네트워크 토폴로지 정보를 기반으로 단순히 최단 경로만을 찾기 때문에 최선 트래픽간에는 부하 분산이 이루어지기 어렵다는 단점이 있다.

반면에 상대 임계치 스킴에서는 라우트 서버가 동적인 네트워크 상태 정보를 부분적으로나마 정기적으로 알려주기 때문에 이를 이용하여 클라이언트는 C_NTDB에 가용 대역폭 정보를 유지할 수 있다. 이와 같은 업데이트 방식으로 인해 C_NTDB는 완전히 정확하게 유지될 수는 없지만, 라우트 서버가 정기적으로 자신이 마지막으로 선정한 링크 상태 정보에서 일정 임계치 이상의 변화가 발생한 링크들에 대하여 새로운 정보를 알려주기 때문에 임계치가 허용하는 범위 내에서 정확한 정보를 유지할 수 있다. 상대 임계치 스킴을 수행하는 클라이언트 라우터는 (그림 3)에서와 같이 변형된 Dijkstra 알고리즘을 이용하여 경로를 계산한다. (그림 3)의 변형된 Dijkstra 알고리즘은 각 송·수신지 쌍에 대하여 존재하는 최단 경로들을 모두 계산하고 이들 각 최단 경로의

가용 대역폭에 비례하는 확률에 의해 무작위로 하나의 경로를 선택한다. 이와 같이 함으로써, 각 클라이언트는 QoS 트래픽을 효과적으로 우회하여 최선 트래픽을 라우팅할 뿐 아니라 최선 트래픽들 간에도 부하를 분산시킨다.

```

modified_dijkstra_best (s, d, b);
begin
M := {s};
for each node n in (V-M) do
D(n) := d(s, n);
While d ⊆ M do
Find a node w in (V - M) such that D(w) is minimum;
if (V - M) = ∅ then
return FALSE;
Add w to M;
if w = d then
return TRUE;
for each node n in (V - M) do
D(n) := Min (D(n), D(w) + d(w, n));
if (D(w)+d(w, n) = D(n)) then
D(n) := Random ( );
return TRUE;
end
Random ( );
begin
select path randomly with probability based on its B(w);
return path;
end
V : 네트워크상에 노드들의 집합
E : 네트워크상에 링크들의 집합
k : 노드의 개수
s : 송신지 노드
d : 수신지 노드
M : s로부터의 최소 비용 경로가 이미 발견된 노드들의 집합
d(i, j) : 노드 i와 j 사이의 비용
d(i, j) = 0
d(i, j) = ∞ 두 노드 사이에 연결된 링크가 없는 경우
D(n) : s로부터의 n노드 까지의 최소 비용 경로의 비용
b : 세션이 요구한 대역폭
B(n) : s로부터의 n노드 까지의 최소 비용 경로의 대역폭
    
```

(그림 3) 상대 임계치 갱신 스킴에서 최선 트래픽간의 부하 분산을 위해 사용하는 변형된 Dijkstra 최단 경로 알고리즘

<표 1>은 절대 임계치 갱신 스킴과 상대 임계치 갱신 스킴의 차이점을 비교, 정리한 것이다.

<표 1> 절대 임계치 갱신과 상대 임계치 갱신의 차이점

기준 \ 스킴	절대 임계치 갱신	상대 임계치 갱신
대상 링크 선정 기준	링크의 활용율이 특정 임계치보다 높은지의 여부	마지막으로 선정한 가용 대역폭에 대한 가용 대역폭 변화 정도가 임계치보다 큰가의 여부
브로드캐스트시 포함되는 정보	해당 링크의 아이디	해당 링크의 아이디와 현재 가용 대역폭
C_NTDB에 유지하는 정보	네트워크 토폴로지 상의 모든 링크들에 대해 존재 유무에 대한 정보	네트워크 토폴로지 상의 모든 링크들에 대해 현재 가용 대역폭에 대한 정보

3. 시뮬레이션 모델

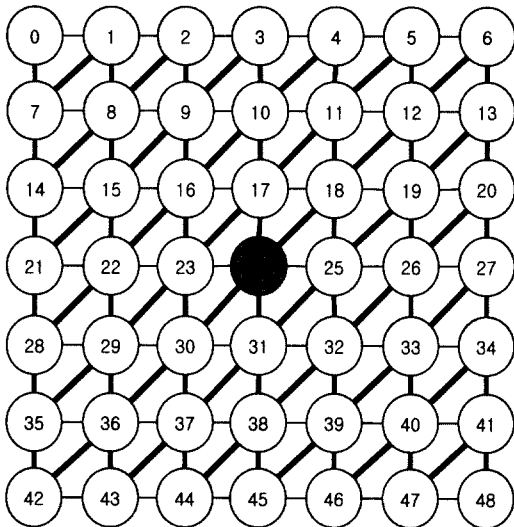
이 장에서는 제안하는 다중 서비스 클래스 지원 라우팅 방안의 성능을 평가하기 위한 시뮬레이션 모델에 대해 설명한다. 시뮬레이션의 효율성을 위하여 다음과 같은 두 가지의 목적에 따라 각각 별도의 시뮬레이션을 수행하였다.

- 제안하는 다중 서비스 클래스 라우팅의 QoS 라우팅 성능과 기존의 QoS 라우팅의 성능 비교
- 제안하는 다중 서비스 클래스 라우팅의 최선 트래픽 라우팅 성능과 기존의 최선 라우팅의 성능 비교

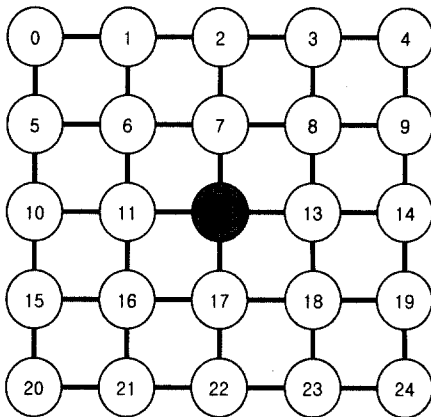
3.3절에서 자세히 설명될 것이지만 QoS 트래픽에 대한 라우팅 성능은 지연이나 손실과 같은 패킷 단위의 성능 측정치가 아닌 세션 수용율과 같은 세션 단위의 성능 측정치로 표시된다. 또한, QoS 트래픽의 경로 결정에서 제안하는 방안이나 기존의 방안 모두 최선 트래픽을 고려하지 않기 때문에 QoS 트래픽에 대한 라우팅 성능 측정을 위해서는 최선 트래픽을 발생시킬 필요가 없다. 따라서 시뮬레이션의 효율을 위하여 QoS 라우팅 성능 측정을 위한 시뮬레이션에서는 QoS 트래픽만을 발생시키고 세션 수준의 사건 운행형(event-driven) 시뮬레이션 방법을 이용하였다. 최선 트래픽의 경우는 지연, 손실 등의 패킷 수준의 측정치로서 라우팅 성능이 측정되고, QoS 트래픽의 분포가 최선 트래픽의 라우팅 성능에 영향을 미치므로 최선 트래픽의 성능을 측정하기 위한 시뮬레이션에서는 QoS 트래픽과 최선 트래픽이 공존하는 환경 하에 패킷 수준의 사건 운행형 시뮬레이션을 수행하였다. 본 시뮬레이션은 C 언어로 구현하였으며, Sun Solaris 워크스테이션에서 수행되었다. 총 시뮬레이션 시간은 세션 및 패킷 수준의 시뮬레이션에서 각각 10800초, 4000초로 하였다.

3.1 네트워크 토폴로지

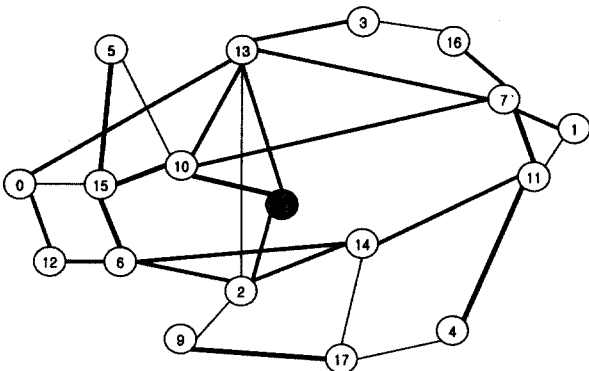
(그림 4)는 시뮬레이션에서 사용된 네트워크 토폴로지 모델을 보여주고 있다. QoS 라우팅 성능 측정을 위한 시뮬레이션에서는 네트워크 토폴로지가 (그림 4)(a)와 같이 49개의 라우터로 구성된 7×7 메쉬 구조인 경우와 (그림 4)(c)와 같이 18개의 라우터로 구성된 일반적인 ISP 구조인 경우를 각각 실험하였다. 이 때 라우트 서버는 각 네트워크 토폴로지 상에서 중앙에 위치한다고 가정하였다. QoS 라우팅 성능 비교를 위한 시뮬레이션 중, 네트워크 크기 변화에 따른 성능 변화를 보기 위한 실험에서는 메쉬의 크기를 3×3, 5×5, 7×7, 9×9, 11×11로 변화시켜 보았다. 각 노드사이의 링크는 양방향 통신을 구분하여 대역폭을 할당하였고 링크의 대역폭은 메쉬 구조에서는 40, 45, 50Mbps 값을 각각 수평방향, 수직방향, 대각선 방향으로 분포시켰으며, ISP 구조에서는 이들 세 가지 값을 임의로 분포시켰다.



(a) 메쉬 기반 구조(QoS)



(b) 메쉬 기반 구조(최선 트래픽)



(c) isp 구조

(그림 4) 시뮬레이션에서 사용한 네트워크 토폴로지

한편, 최선 트래픽의 라우팅 성능을 측정하기 위한 시뮬레이션에서는 패킷 수준의 시뮬레이션으로 인해 QoS 라우팅 성능 측정을 위한 시뮬레이션과 같은 정도의 트래픽을 발생시킨다면 지나치게 시뮬레이션 시간이 길어지므로 적정

수준의 시간 내에 시뮬레이션을 수행하기 위하여 트래픽 발생율을 낮추고 네트워크 용량도 상대적으로 낮은 경우를 실험하였다. 네트워크 토폴로지가 (그림 4)(b)와 같이 25개의 라우터로 구성된 5×5 메쉬인 경우와 (그림 4)(c)와 같이 18개의 라우터로 구성된 일반적인 ISP 구조인 경우를 각각 실험하였는데, 각 노드사이의 링크는 역시 양방향 통신을 구분하여 대역폭을 할당하였고 링크의 대역폭은 메쉬 구조에서는 모든 링크를 10Mbps로 설정하였고, ISP 구조에서는 8Mbps, 10Mbps, 12Mbps의 값을 임의로 분포시켰다. 모든 경우, 각 링크의 전파 지연은 0.5ms라고 가정하였으며, 시뮬레이션 기간 중 링크 실패(failure)는 없다고 가정하였다.

3.2 트래픽 모델

본 시뮬레이션의 목적은 제안하는 스킴 및 비교 대상 스킴들의 대체적인 경향 및 성능 비교를 제시하는 것이므로 데이터 트래픽의 특성을 좀 더 실제적으로 모델링하는 복잡한 트래픽 모델을 피하고 이와 같은 목적을 만족시키기에는 충분한 기본적인 단순한 트래픽 모델을 사용하였다. QoS 라우팅 성능 비교를 위한 시뮬레이션에서는 각 노드에서 독립적으로 포아송(Poisson) 분포에 따라 세션 요청이 발생한다고 가정하였고, 네트워크 트래픽 부하를 변화시키기 위하여 QoS 트래픽의 세션 발생 간격을 변화시켰다. 한 QoS 세션이 요구하는 대역폭은 (64Kbps, 6Mbps)의 범위에서 균일 분포(Uniform distribution)에 따라 정해지도록 하였고, 세션 길이는 지수 분포의 랜덤 변수(exponentially distributed random variable)이며 평균 세션 길이는 180초라고 가정하였다. 네트워크 크기 변화에 따른 성능 변화에 관한 실험에서는 모든 크기의 메쉬에 대해 하나의 클라이언트 라우터에서 발생하는 세션의 생성률은 동일하도록 조정하였다. 또 전체 트래픽 중 메쉬에서는 90%, ISP에서는 80%가 송·수신지 쌍들 중 무작위로 뽑은 10%의 송·수신지 쌍들간에 발생하도록 하였다.

최선 트래픽의 성능 측정을 위한 시뮬레이션에서는 최선 트래픽과 QoS 트래픽을 모두 발생시키되, 시뮬레이션 시간의 단축을 위하여 최선 트래픽은 패킷 수준으로 다루고 QoS 트래픽은 세션 수준으로 다루었다. 즉, QoS 세션의 경우는 실제로 패킷을 발생시키지 않고, 세션이 경유하는 경로 상의 링크들의 가용 대역폭을 QoS 세션이 요구하는 대역폭만큼 감소시켰다. QoS 라우팅 성능에 관한 실험에서와 같이 각 노드에서의 세션 발생은 QoS 세션이나 최선 트래픽 세션에 관계 없이 독립적으로 이루어지며 포아송 분포를 따른다고 가정하였고 트래픽 부하를 변화시키기 위해 세션 발생 간격을 변화시켰다. 그리고, QoS 세션과 최선 트래픽 세션의 발생 비율을 3 : 1, 1 : 1, 1 : 3 세 가지로 변화시켜 보았다. 하나의 QoS 세션이 요구하는 대역폭은 2Mbps로 고정시켰으며, 세션 길이는 지수 분포 랜덤 변수로 QoS 세션의 경

우는 평균 세션 길이가 180초이고 최선 트래픽 세션의 경우는 90초라 가정하였다. 또한 최선 트래픽의 경우, 세션 내에서의 패킷 발생 간격은 평균이 0.8초인 지수 분포를 따른다고 가정하였고 패킷 하나의 크기는 1024바이트라 가정하였다.

3.3 성능 측정 메트릭

QoS 라우팅의 성능은 대역폭 수용율 *BAR*(Bandwidth Acceptance Ratio)로 측정하였는데, *BAR*은 다음과 같이 정의된다[10].

$$BAR = \frac{\sum_{i \in A} bw(i)}{\sum_{i \in R} bw(i)}$$

여기서 *R*는 QoS 경로를 요청하는 모든 세션의 집합을, *A*는 QoS 경로 요청이 수용된 세션의 집합을 말하며, *bw(i)*는 세션 *i*가 요구하는 대역폭을 의미한다.

QoS 라우팅을 위한 프로토콜 오버헤드는 크게 다음과 같이 두 가지를 분석하였다: (a) 네트워크 대역폭 소모(*NBC*: Network Bandwidth Consumption), (b) 제안하는 라우팅 방안에서의 라우트 서버 혹은 기존의 QoS 라우팅에서 임의의 한 라우터가 부담하는 메시지 프로세싱 오버헤드(*MPO*: Message Processing Overhead). *NBC*는 총 시뮬레이션 시간 동안 각 링크에서 전송된 모든 프로토콜 제어 메시지의 합으로 측정하였는데, 제안하는 서버 기반 QoS 라우팅의 경우는 2장에서 설명한 제어 메시지들의 수를 세었고, 기존의 QoS 라우팅 스킴의 경우는 네트워크 상태 정보를 전달하기 위해 플러딩되는 제어 메시지들의 수를 세었다. *MPO*는 제안하는 라우팅 방안의 경우는 라우트 서버에서 전송하거나 전송 받은 메시지 수를 세었고, 이와 비교된 분산 QoS 라우팅의 경우는 임의의 라우터 하나에서 전송하거나 전송 받은 메시지 수의 평균을 구하였다.

최선 라우팅의 성능은 최선 패킷의 평균 처리량 *AT*(Average Throughput), 평균 지연 *AD*(Average Delay), 및 손실을 *PLR*(Packet Loss Rate)로서 측정하였고, 이들은 각각 다음과 같이 정의된다[11].

$$AT = \frac{\sum_{i \in B} transmitted\ bits(i)}{\sum_{i \in B} duration(i)}, \quad AD = \frac{\sum_{i \in T} delay(i)}{n | T |},$$

$$PLR = \frac{n | L |}{n | G |}$$

여기서 *B*는 모든 최선 트래픽 세션의 집합을, *G*는 총 발생된 패킷의 집합을, *T*는 수신지까지 제대로 전송된 패킷의 집합을, *L*는 손실된 패킷의 집합을 말한다. 제안하는 다중 서비스 클래스 라우팅에서는 최선 트래픽의 성능 향상을 위하여 정기적으로 라우트 서버가 제어 메시지 LSU를 플러

딩하는 오버헤드가 발생한다. 그러나, 라우트 서버만이 LSU를 플러딩하기 때문에 기존의 QoS 라우팅에서 모든 라우터들이 네트워크 상태 정보를 플러딩하는 것에 비하여 플러딩 오버헤드가 무시해도 좋을 만큼 극히 적을 뿐 아니라 플러딩 인터벌과 네트워크 규모에 따라 그 양이 고정적이기 때문에 최선 트래픽 성능을 측정하는 시뮬레이션에서 이를 측정하지는 않았다.

4. 시뮬레이션 결과 분석

4.1 QoS 라우팅 성능

(그림 5)~(그림 7)은 제안하는 서버 기반 QoS 라우팅의 경로 캐싱 스킴 가운데 1-경로캐싱인 경우와 기존에 제안된 분산 QoS 라우팅 스킴들과의 성능을 비교한 것이다. 비교 대상이 된 기존의 분산 QoS 라우팅 스킴들은 현재 인터넷에서 가장 널리 사용되고 있는 라우팅 방식인 OSPF를 확장한 대표적인 QoS 라우팅 스킴들이다. 이들 방식은 임계치 기반 및 지수 클래스 기반 네트워크 상태 정보 갱신 정책을 사용하는 스킴들로 구분된다[9]. 임계치 기반 방식은 가용 대역폭 값이 마지막으로 선정한 가용 대역폭 값의 일정 비율(임계치) 이상으로 변하면 다시 해당 링크에 대한 가용 대역폭 정보를 선전하는 방식이다. 한편, 지수 클래스 기반 방식은 가용 대역폭 값을 몇 개의 클래스로 분류하고 가용 대역폭 값이 클래스 경계를 넘어 변화하게 되면 다시 해당 링크의 가용 대역폭 값을 선전하는 방식인데, 각 클래스의 범위는 '기본 클래스 크기' 및 '클래스 성장 파라미터'라는 두 파라미터에 의하여 결정된다. 큰 가용 대역폭 값일수록 하나의 클래스에 해당하는 가용 대역폭 값의 범위(클래스의 최대 가용 대역폭 값 - 클래스의 최소 가용 대역폭 값)가 더 커지게 되는데, '기본 클래스 크기'는 가장 작은 크기의 클래스의 크기로서 하나의 QoS 플로우가 요구할 수 있는 최대 대역폭 요구량의 몇 %인가로 표시되고, '클래스 성장 파라미터'는 임의의 클래스의 범위가 그 전 단계 클래스 범위의 몇 배인지를 의미한다. 본 논문의 시뮬레이션에서는 기존의 분산 QoS 라우팅 스킴들의 파라미터 값에 대하여 이들 정책에 관한 연구에서 가장 일반적으로 권고하는 값들을 사용하였다. 임계치 기반 갱신 정책의 경우 임계치 값으로 60%를, 지수 클래스 기반 갱신 정책의 경우 기본 클래스 크기 및 클래스 성장 파라미터로 각각 200%와 2를 사용하였다. 뿐만 아니라 임계치 및 지수 클래스 기반 갱신 정책과 클램프 다운(clamp down) 타이머 정책을 조합한 경우도 비교하여 보았다. 클램프 다운 타이머는 가용 대역폭 선전 오버헤드를 제어하기 위하여 선전 후 일정 시간(클램프 다운 타임)이 경과한 경우에만 새로운 선전을 할 수 있도록 하는 방법이다.

(그림 5), (그림 6), (그림 7)에서의 라인들은 다음과 같이

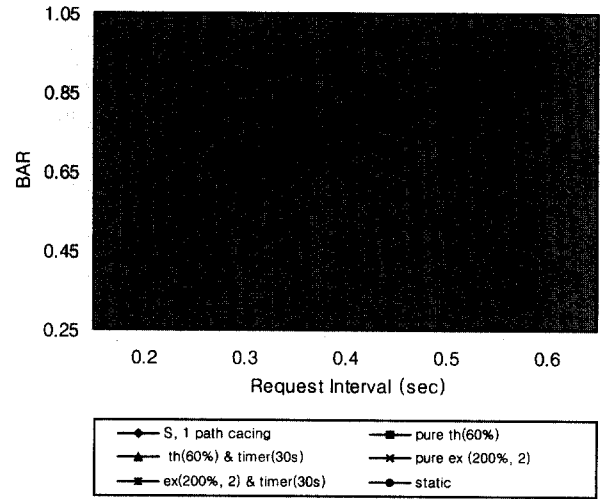
각각의 스킴에 해당하는 결과를 보여주고 있다.

- S, 1-path caching : 제안하는 서버기반 1-경로캐싱 스킴
- pure th(60%) : 60%를 임계치로 하는 순수 임계치 기반 갱신 정책[9]
- th(60%)&timer(30s) : 30초 크기의 클램프 다운 타이머를 사용하고 60%를 임계치로 하는 임계치 기반 갱신 정책[9]
- pure ex(200%, 2) : '기본 클래스 크기'가 200%이고 '클래스 성장 파라미터'가 2인 순수 지수 클래스 기반 갱신 정책[9]
- ex(200%, 2)&timer(30s) : 30초 크기의 클램프 다운 타이머를 사용하고 '기본 클래스 크기'가 200%이고 '클래스 성장 파라미터'가 2인 지수 클래스 기반 갱신 정책[9]
- static : 정적 라우팅

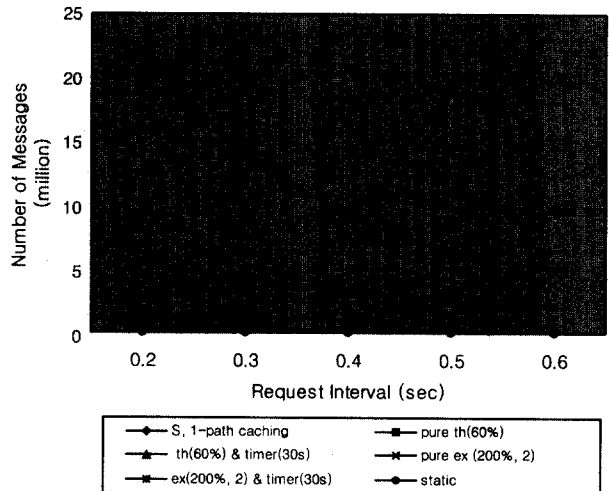
(그림 5), (그림 6)은 각각 메쉬와 ISP 구조에서 네트워크에 추가되는 트래픽 부하 변화에 따른 성능 변화를 보여준 것이다. 이들 그래프의 x축은 세션 요청 발생 간격의 평균을 나타내는데, 세션 요청 발생 간격이 짧아짐에 따라 트래픽 부하가 커지게 된다. (그림 7)은 네트워크 크기 변화에 따른 각 스킴들의 라우팅 성능 변화를 보여준 것이다. 이 실험에서는 각 노드에서의 세션 발생 간격을 동일하게 유지하면서 전체 클라이언트 노드 수를 9개에서 121개까지 변화시켜 보았다.

라우팅 성능을 표시하는 BAR을 비교해 보면 모든 경우에 대하여 제안하는 방안이 기존의 분산 QoS 라우팅 방안들에 비해 우수하다((그림 5)(a), (그림 6)(a), (그림 7)(a) 참조). 네트워크의 대역폭 소모량(NBC) 면에서 보면 제안하는 방안은 기존 방안들 중 클램프 다운 타이머 정책과 조합한 상태 정보 갱신 방안들과 유사한 정도의 오버헤드를 발생시키며, 기존 분산 QoS 라우팅 방안들 중 가장 라우팅 성능이 높은 순수 임계치 기반 갱신 정책보다는 훨씬 낮은 대역폭 소모 오버헤드를 발생시킨다((그림 5)(b), (그림 6)(b), (그림 7)(b) 참조). 또한, 기존의 분산 QoS 라우팅 방안 가운데 가장 성능이 좋은 임계치 기반 갱신 정책을 사용하는 경우의 임의의 한 라우터보다는 제안하는 방안의 라우터 서버가 훨씬 낮은 메시지 프로세싱 오버헤드(MPO)를 부담함을 볼 수 있다((그림 5)(c), (그림 6)(c), (그림 7)(c) 참조).

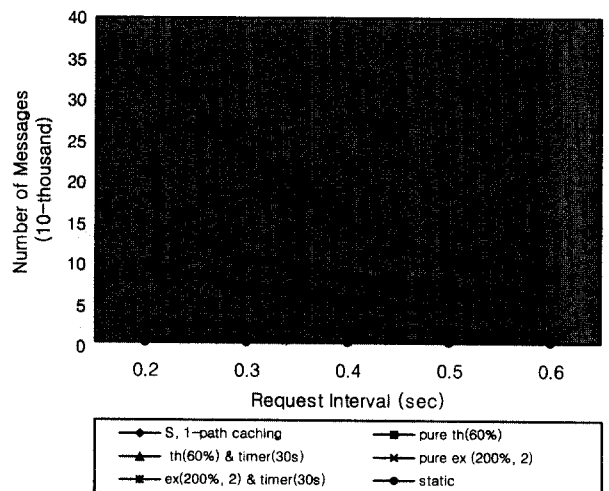
네트워크 규모의 변화에 따른 성능 변화를 비교해 보면((그림 7) 참조), 실험한 모든 경우에 대하여 제안하는 라우팅 방안의 BAR이 가장 높을 뿐 아니라 다른 기존의 분산 QoS 라우팅 방안들의 경우 네트워크 규모가 커짐에 따라 BAR이 감소하는데 반해 제안하는 라우팅 방안은 거의 일정한 BAR을 유지함을 볼 수 있다. 기존의 분산 QoS 라우팅



(a) 라우팅 성능(BAR)

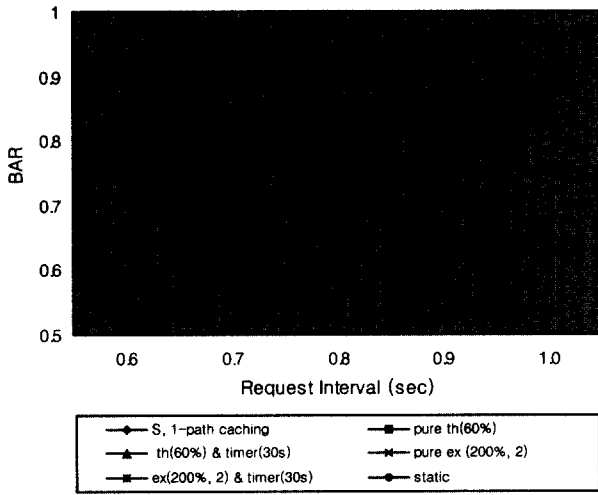


(b) 네트워크 대역폭 소비(NBC)

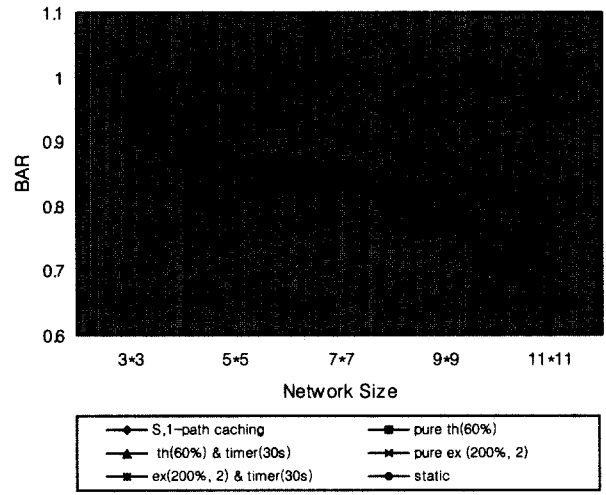


(c) 라우트 서버/라우터 당 메시지 프로세싱 오버헤드(MPO)

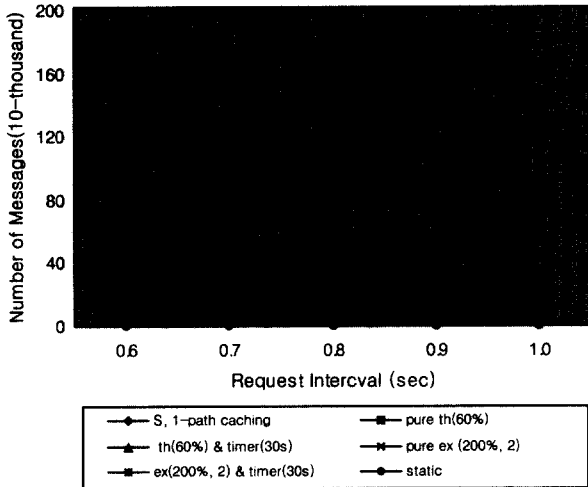
(그림 5) 메쉬 구조에서의 트래픽 부하 변화에 따른 QoS 라우팅 성능 및 오버헤드 변화



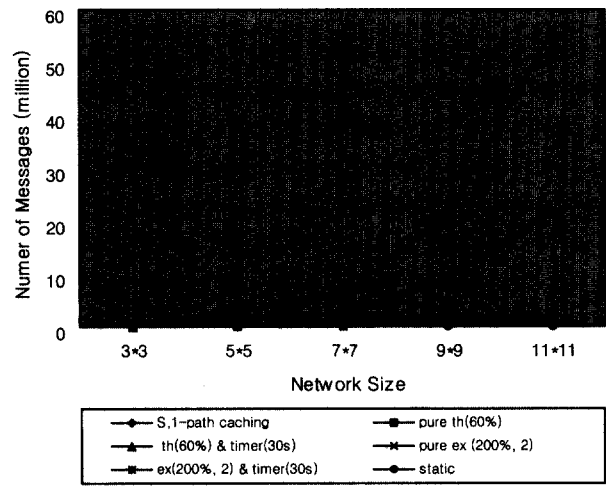
(a) 라우팅 성능(BAR)



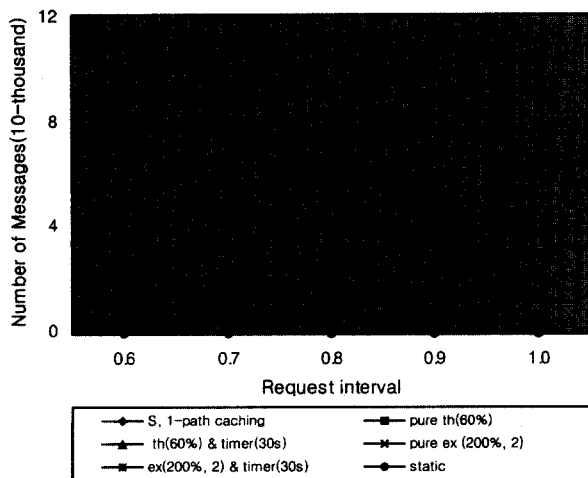
(a) 라우팅 성능(BAR)



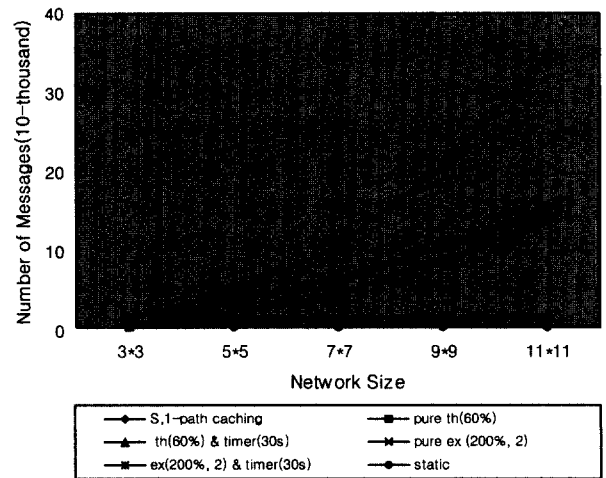
(b) 네트워크 대역폭 소비(NBC)



(b) 네트워크 대역폭 소비(NBC)



(c) 라우트 서버/라우터 당 메시지 프로세싱 오버헤드(MPO)
(그림 6) ISP 네트워크 구조에서의 트래픽 부하 변화에 따른 QoS 라우팅 성능 및 오버헤드 변화



(c) 라우트 서버/라우터 당 메시지 프로세싱 오버헤드(MPO)
(그림 7) 네트워크 규모 변화에 따른 QoS 라우팅 성능 및 오버헤드 변화

방안들은 각 라우터들이 독자적으로 라우팅 결정을 내리므로 네트워크 크기가 커짐에 따라 이들의 분산된 결정 간의 상호 모순 정도가 더 커지게 되고 이로 인해 라우팅 성능이 떨어지게 된다. (그림 7)(b)의 네트워크 대역폭 소비에 관한 결과를 보면 순수 임계치 기반 갱신 정책 및 지수 클래스 기반 갱신 정책의 경우는 네트워크 크기가 커짐에 따라 NBC가 급격하게 증가하는데 반해, 제안하는 방안은 타이머 기반 갱신 정책의 분산 QoS 라우팅의 경우와 비슷한 정도로 오버헤드가 증가함을 알 수 있다. (그림 7)(c)를 보면 제안하는 방안에서 라우트 서버의 메시지 프로세싱 오버헤드는 순수 지수 기반 갱신 정책과 유사한 정도이고 네트워크 크기가 확장됨에 따라 다소 증가하는 형태를 갖는다. 그러나 기존 분산 QoS 라우팅 방안들 중 가장 높은 라우팅 성능을 갖는 순수 임계치 기반 갱신 정책의 경우 네트워크 크기가 커짐에 따라 이 오버헤드가 훨씬 더 급격하게 증가함을 볼 수 있다.

4.2 최선 트래픽의 라우팅 성능

QoS 트래픽과 최선 트래픽의 비율을 3:1, 1:1, 1:3으로 변화시켜 보면서 메쉬 구조의 네트워크와 일반적인 ISP 구조의 네트워크에 대하여 모두 제안하는 최선 라우팅 스킴들과 기존의 정적 최선 라우팅 스킴의 성능을 비교하였다. [11, 12]의 라우팅 스킴들은 목적 면에서 본 논문에서 제안하는 최선 트래픽 라우팅 스킴과 동일하지만 이들 스킴과 본 논문에서 제안하는 최선 트래픽 라우팅 스킴은 그 기반 가정이 되는 QoS 라우팅 방식이 서로 달라 서로 비교하기 어려우므로, 제안하는 스킴의 특성 및 성능 향상 정도를 보기 위하여 제안하는 스킴이 기반으로 하고 있는 QoS 라우팅 방식 하에 제안하는 스킴이 적용되는 경우와 그렇지 않은 경우를 비교한 것이다. (그림 8)~(그림 10)은 메쉬 구조 네트워크의 경우에 대한 결과이고, (그림 11)~(그림 13)은 ISP 구조 네트워크에 대한 결과이다. 각 그림에서의 라인들은 다음과 같은 각각의 스킴에 해당하는 결과를 보여주고 있다.

- 절대(x%) : x%를 임계치로 하는 절대 임계치 갱신 스킴
- 상대(y%) : y%를 임계치로 하는 상대 임계치 갱신 스킴
- static : 정적 최선 라우팅

상대 임계치 갱신의 경우 분산 QoS 라우팅의 임계치 갱신 정책을 참조하였는데[9], 파라미터 값에 대하여 이들 정책에 관한 연구에서 가장 일반적으로 권고하는 값인 60%를 기본적으로 사용하였다. 또한 임계치의 변화는 60%를 기준으로 더 작은 임계치와 더 큰 임계치로 각각 20%와 100%를 선정하였다.

또한 각 그림의 (a), (b), (c)는 트래픽 부하 변화에 따른 AT(Average Throughput), PLR(Packet Loss Rate) 및 AD(Average Delay)를 나타낸다. 네트워크의 트래픽 부하는

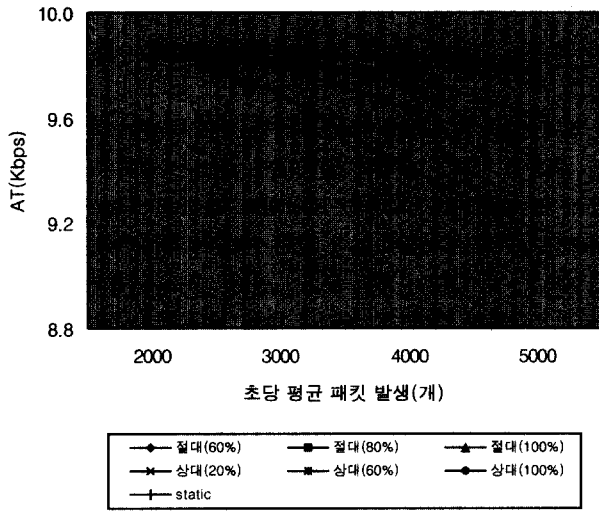
초당 발생하는 평균 패킷 개수로 조절하였는데, 이 때의 패킷 개수는 QoS 트래픽과 최선 트래픽 패킷의 수를 합한 것이다.

먼저 QoS 트래픽과 최선 트래픽 양의 비율이 최선 트래픽 성능에 미치는 영향을 살펴보면, QoS 트래픽 부하의 비율이 높아질수록 직관적인 예상대로 라우팅 스킴이나 네트워크 토폴로지에 관계 없이 모두 최선 트래픽의 성능이 전반적으로 낮아지는 것을 볼 수 있다. 그러나 유의할 것은 QoS 트래픽의 양이 더 많거나 QoS 트래픽과 최선 트래픽의 양이 같은 정도인 경우((그림 8), (그림 9)와 (그림 11), (그림 12))에는 제안하는 라우팅 방안에 의하여 최선 트래픽의 성능이 향상된다는 것이다. 단, QoS 트래픽의 양이 상대적으로 훨씬 적은 경우에는 기존의 최선 트래픽 라우팅을 사용하는 경우와 제안하는 라우팅 방안을 사용하는 경우 거의 차이가 없음을 볼 수 있다((그림 10)과 (그림 13)).

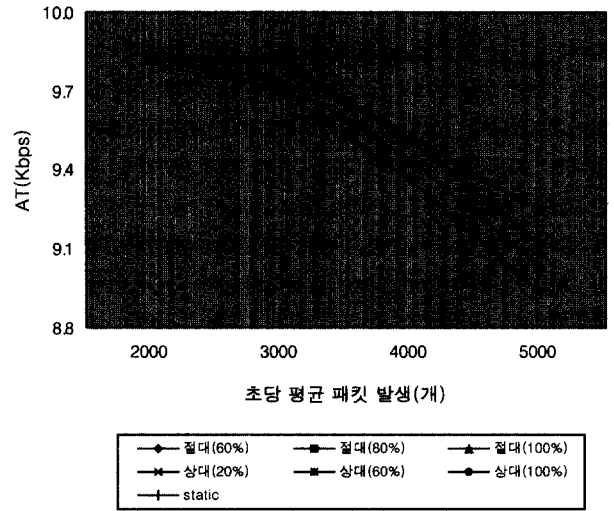
QoS 트래픽이 전체 트래픽의 50% 이상을 차지하는 경우, 제안하는 스킴들은 평균 처리율(AT)과 패킷 손실 비율(PLR) 면에서 모두 기존의 최선 라우팅 스킴보다 우수하다(그림 8)(a), (그림 8)(b), (그림 9)(a), (그림 9)(b), (그림 11)(a), (그림 11)(b), (그림 12)(a), (그림 12)(b) 참조). 특히, 상대 임계치 갱신 스킴의 경우는 여러 개의 최단 경로로 부하를 분산시킴으로써 부하 증가에 따른 성능 감소가 거의 없음을 볼 수 있다. 단, 절대 임계치 갱신에서는 임계치가 60%($U_i = 60\%$ 이상인 링크를 제거한 후 최단 경로 라우팅 수행)일때, QoS 트래픽 비율이 높고 트래픽 부하도 높은 경우 정적 최선 라우팅보다 더 낮은 성능치를 보이는 경우가 있는데, 이것은 트래픽 부하가 높을 때 QoS 트래픽에 의해 60% 이상이 이용되고 있는 링크가 증가하여 다수의 링크 제거로 인해 최선 트래픽을 위해 가용한 경로가 없는 경우가 증가하기 때문이다.

평균 지연(AD)의 경우 절대 임계치 갱신 스킴은 기존의 최선 라우팅과 유사한 수준이고 트래픽 부하가 증가함에 따라 증가하는 추세를 보인다. 그러나 상대 임계치 갱신 스킴은 절대 임계치 갱신 스킴이나 기존의 최선 라우팅 방안보다 AD가 낮을 뿐 아니라 트래픽 부하가 증가해도 AD가 거의 증가하지 않음을 볼 수 있다(그림 8)(c), (그림 9)(c), (그림 11)(c), (그림 12)(c) 참조).

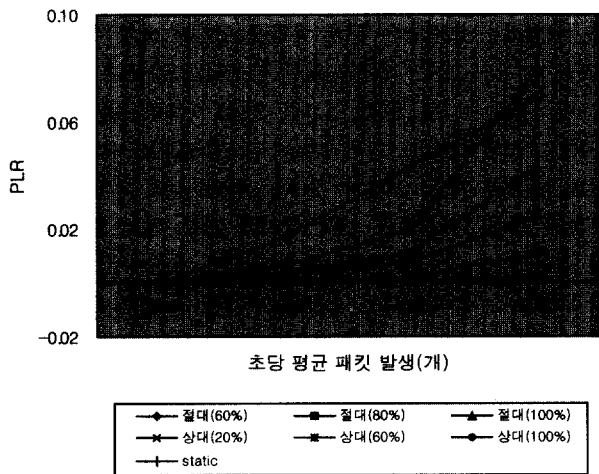
(그림 8)~(그림 13)의 결과를 통하여 제안하는 최선 라우팅 개선 방안인 절대 임계치 갱신 스킴과 상대 임계치 갱신 스킴을 종합적으로 비교해 보면 상대 임계치 갱신 스킴의 성능이 더 우수하다. 또한, 절대 임계치 갱신 스킴은 라우팅 성능이 임계치 설정에 따라 민감하게 영향을 받는데 반해 상대 임계치 갱신 스킴은 실험해 본 범위의 임계치 값에 대해서는 임계치 값에 따른 성능 차이가 매우 적었다. 그리고, 트래픽 부하 증가에 대해서도 상대 임계치 갱신이 더 안정된 성능을 보여주었다.



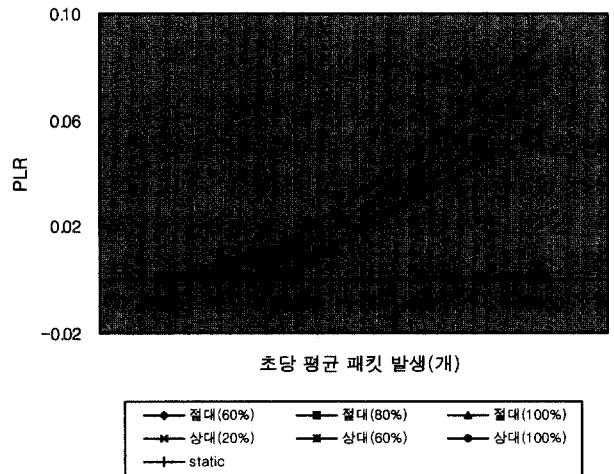
(a) 최선 트래픽의 AT(Kbps)



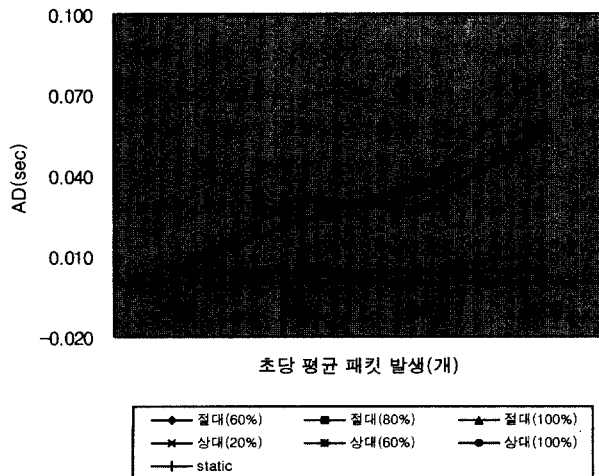
(a) 최선 트래픽의 AT(Kbps)



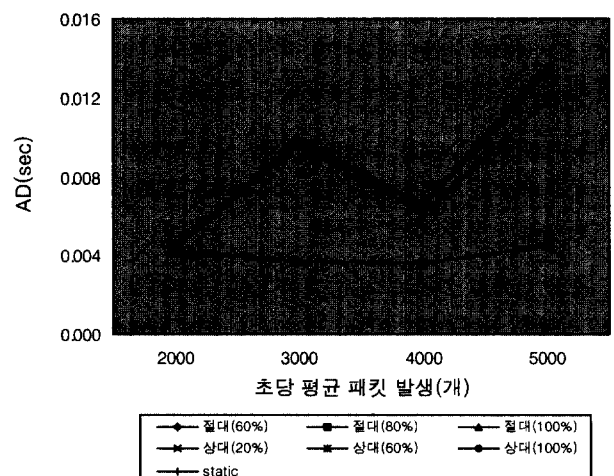
(b) 최선 트래픽의 PLR



(b) 최선 트래픽의 PLR



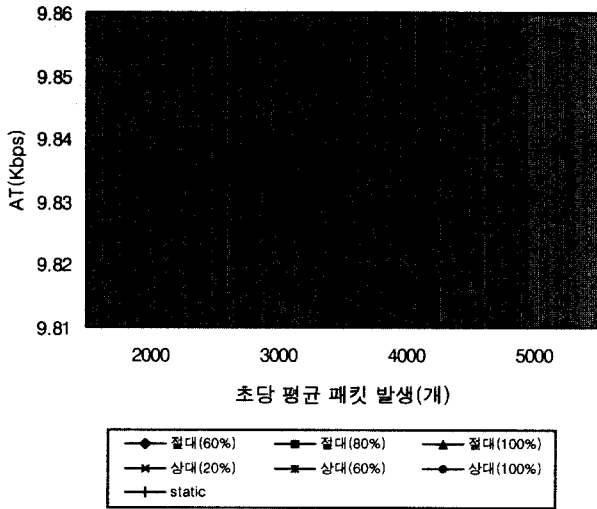
(c) 최선 트래픽의 AD(sec)



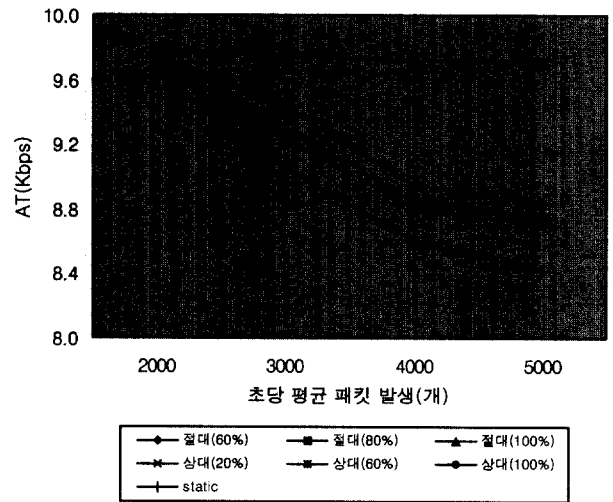
(c) 최선 트래픽의 AD(sec)

(그림 8) 메쉬 구조에서의 트래픽 부하 변화에 따른 최선 트래픽 성능 비교(QoS 트래픽 75%, 최선 트래픽 25%)

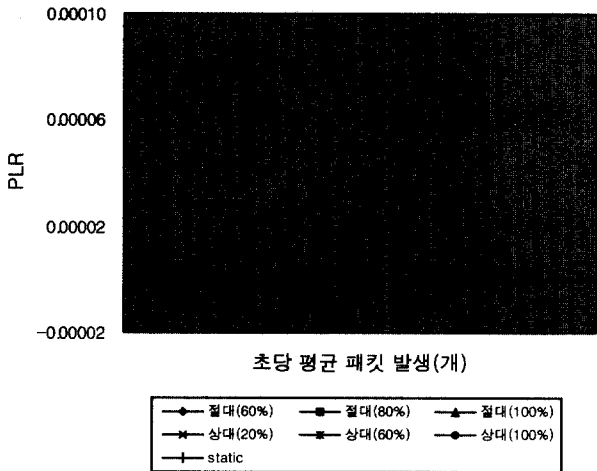
(그림 9) 메쉬 구조에서의 트래픽 부하 변화에 따른 최선 트래픽 성능 비교(QoS 트래픽 50%, 최선 트래픽 50%)



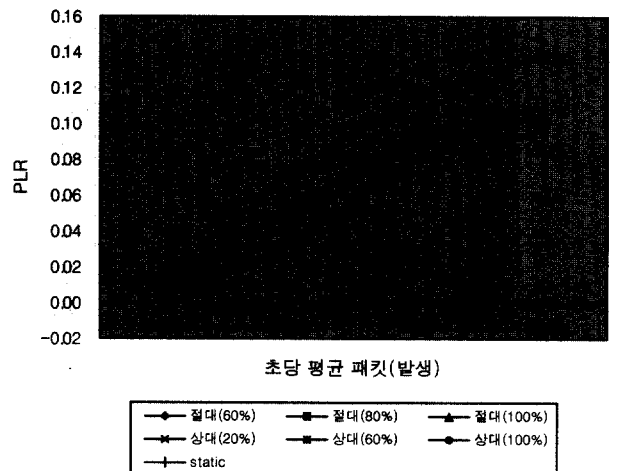
(a) 최선 트래픽의 AT(Kbps)



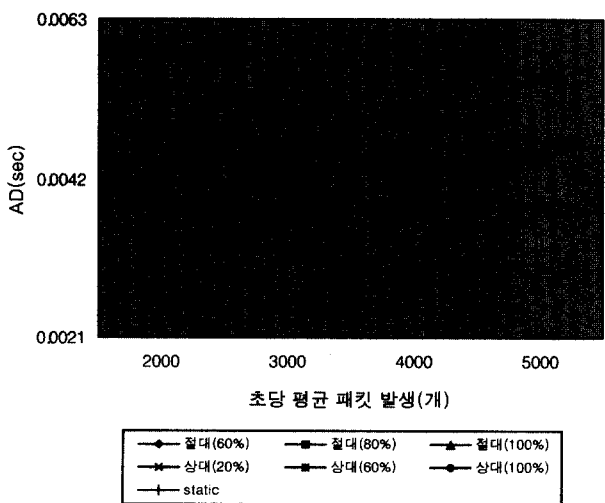
(a) 최선 트래픽의 AT(Kbps)



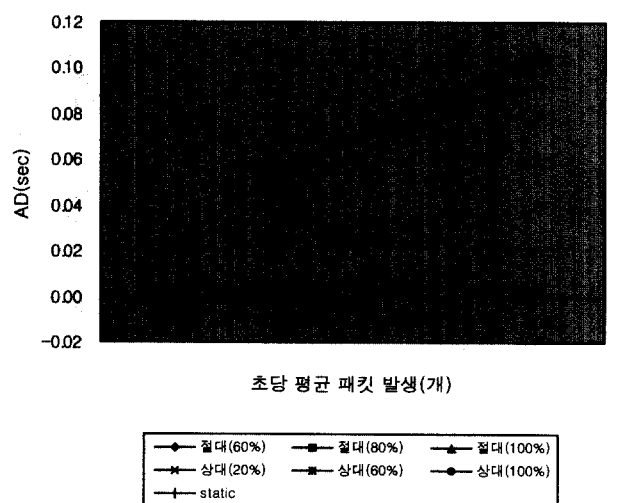
(b) 최선 트래픽의 PLR



(b) 최선 트래픽의 PLR



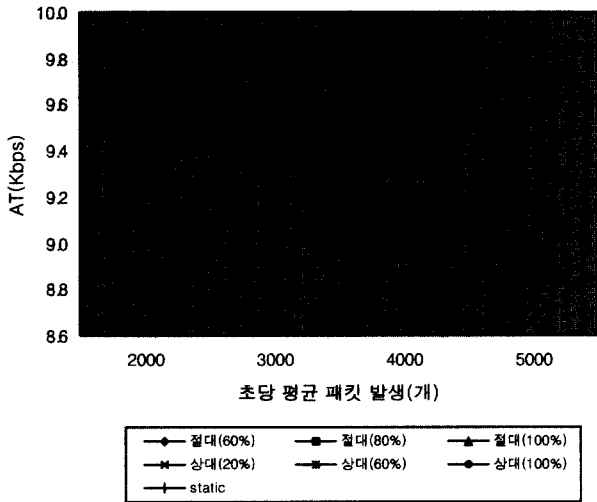
(c) 최선 트래픽의 AD(sec)



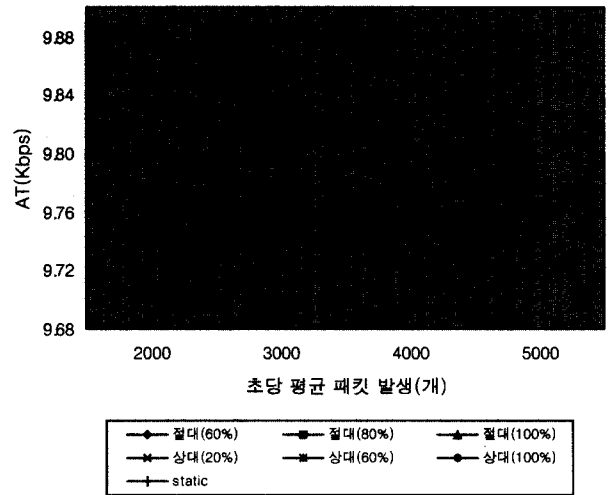
(c) 최선 트래픽의 AD(sec)

(그림 10) 메쉬 구조에서의 트래픽 부하 변화에 따른 최선 트래픽 성능 비교(QoS 트래픽 25%, 최선 트래픽 75%)

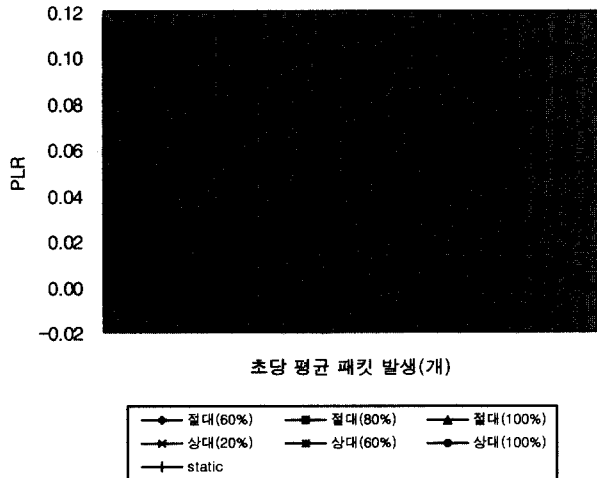
(그림 11) ISP 구조에서의 트래픽 부하 변화에 따른 최선 트래픽 성능 비교(QoS 트래픽 75%, 최선 트래픽 25%)



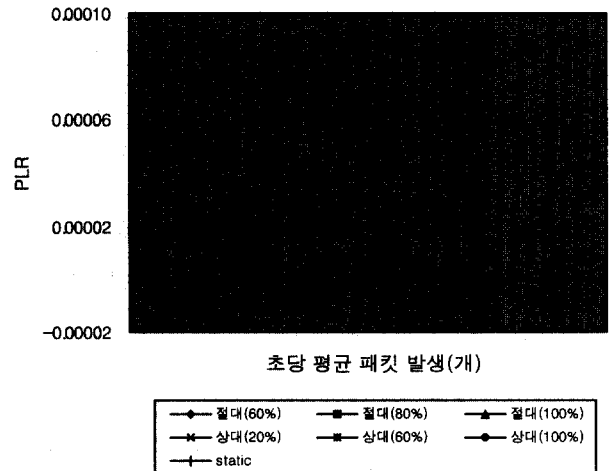
(a) 최선 트래픽의 AT(Kbps)



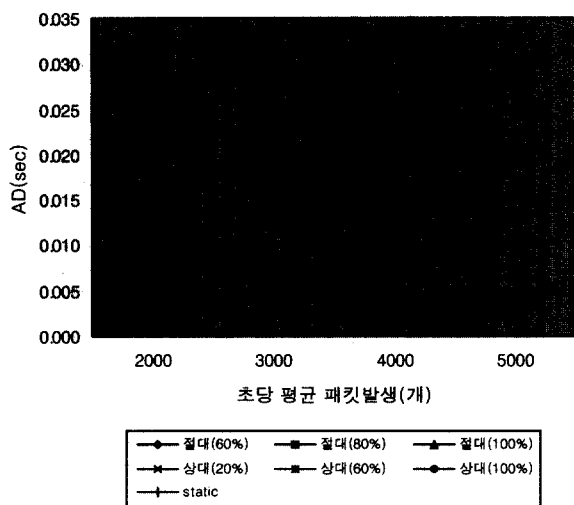
(a) 최선 트래픽의 AT(Kbps)



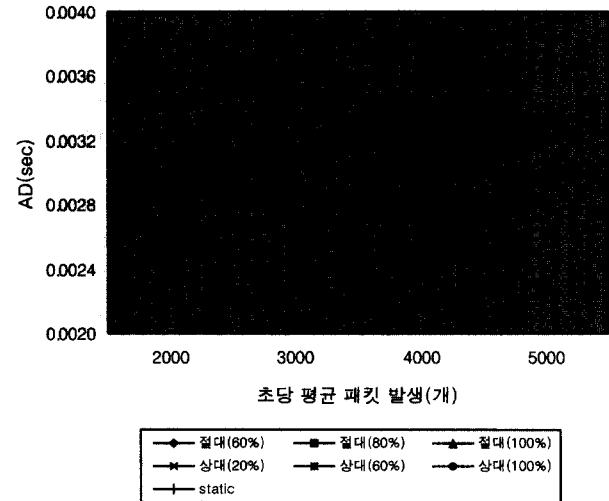
(b) 최선 트래픽의 PLR



(b) 최선 트래픽의 PLR



(c) 최선 트래픽의 AD(sec)



(c) 최선 트래픽의 AD(sec)

(그림 12) ISP 구조에서의 트래픽 부하 변화에 따른 최선 트래픽 성능 비교(QoS 트래픽 50%, 최선 트래픽 50%)

(그림 13) ISP 구조에서의 트래픽 부하 변화에 따른 최선 트래픽 성능 비교(QoS 트래픽 25%, 최선 트래픽 75%)

5. 결 론

본 논문에서는 QoS 라우팅을 효율적으로 지원하면서 최선 트래픽의 성능 또한 향상시킬 수 있는 서버 기반의 라우팅 방안을 제안하였다. 제안하는 라우팅 방안은 크게 다음과 같은 두 가지의 의미를 갖는다.

- 네트워크 상태 정보 교환을 위한 프로토콜 오버헤드가 없는 QoS 라우팅 방안 제시
- QoS 라우팅이 도입된 네트워크에서의 최선 트래픽 클래스의 라우팅 성능 향상 방안 제시

제안하는 스킴에서 라우트 서버는 QoS 경로를 할당하고 반환 받는 작업을 통해 스스로 네트워크 상태 정보를 정확히 파악할 수 있으므로 기존의 QoS 라우팅에서와 같은 네트워크 상태 정보 교환을 위한 프로토콜 오버헤드가 없고, 항상 정확한 네트워크 상태 정보를 기반으로 라우팅 결정을 내리므로 라우팅 성능을 크게 향상시킨다. 뿐만 아니라 라우트 서버가 QoS 트래픽으로 인해 현재 부하가 높은 링크의 상태 정보를 클라이언트 라우터들에게 알려 최선 트래픽 라우팅시 QoS 트래픽을 효과적으로 우회하게 함에 따라 QoS 라우팅 성능에는 전혀 영향을 주지 않고 최선 트래픽의 성능 또한 향상시킬 수 있다.

시뮬레이션 결과, 제안하는 서버 기반 QoS 라우팅이 기존의 QoS 라우팅 스킴들에 비해서 높은 라우팅 성능을 보이는 동시에 프로토콜 오버헤드 측면에서도 매우 우수함을 알 수 있었다. 또 제안하는 최선 라우팅 스킴들은 QoS 트래픽의 비율이 반 이상이 되는 네트워크에서는 기존의 최선 라우팅에 비하여 최선 트래픽의 성능을 향상시킴을 볼 수 있었고, 특히 상대 임계치 갱신 스킴은 효과적으로 QoS 트래픽을 우회할 뿐 아니라 최선 트래픽 간에도 부하를 분산함으로써 트래픽 부하 증가에 효율적으로 대처할 수 있음을 보았다.

제안하는 서버 기반 QoS 라우팅 방안은 단일 서버를 사용하는 구조에서 일반적으로 겪게 되는 단일 지점에서의 병목이나 실패와 같은 문제점들을 가지고 있다. 본 논문에서는 이와 같은 문제는 다루지 않고, 향후 연구로서 제안하는 방안을 다중 라우트 서버 구조로 확장할 것이다.

참 고 문 헌

[1] Q. Ma, P. Steenkiste, "On Path Selection for Traffic with Bandwidth Guarantees," Proc. of IEEE ICNP, 1997.
 [2] R. Guerin, A. Orda, D. Williams, "QoS Routing Mechanisms and OSPF Extensions," Proc. of IEEE GLOBECOM 1997.
 [3] J. -Y. Le Boudec, T. Przygienda, "A Route Precomputation Algorithm for Integrated Services Networks," Jour-

nal of Networks and Systems Management, Vol.3, No.4, pp.427-449, 1995.
 [4] A. Shaikh, J. Rexford, K. Shin, "Efficient Precomputation of Quality-of-Service Routes," Proc. of Workshop on Network and Operating Systems Support for Digital Audio and Video, 1998.
 [5] G. Apostolopoulos, S. K. Tripathi, "On the Effectiveness of Path Pre-Computation in Reducing the Processing Cost of On-Demand QoS Path Computation," Proc. of IEEE INFOCOM 1998.
 [6] G. Apostolopoulos, R. Guerin, S. Kamat, S. K. Tripathi, "On Reducing the Processing Cost of On-Demand QoS Path Computation," Journal of High Speed Networking, Vol.7, No.2, pp.77-98, 1998.
 [7] M. Peyravian, A. D. Kshemkalyani, "Network Path Caching : Issues, Algorithms, and a Simulation Study," Computer Communications, Vol.20, pp.605-614, 1997.
 [8] A. Shaikh, J. Rexford, K. Shin, "Dynamics of quality-of-service routing with inaccurate link-state information," University of Michigan Technical Report CSE-TR-350-97, 1998.
 [9] G. Apostolopoulos, R. Guerin, S. Kamat, S. K. Tripathi, "QoS Routing : A Performance Perspective," Proc. of ACM SIGCOMM, 1998.
 [10] G. Apostolopoulos, R. Guerin, S. Kamat, S. K. Tripathi, "Server based QoS Routing," Proc. of IEEE GLOBECOM, 1999.
 [11] Q. Ma, P. Steenkiste, "Supporting Dynamic Inter-class Resource Sharing : A Multi-Class QoS Routing Algorithm," Proc. of IEEE INFOCOM, 1999.
 [12] H. Kochkar, T. Ikenaga, Y. Oie, "QoS Routing Algorithm Based on Multiclasses Traffic Load," Proc. of IEEE GLOBECOM, 2001.
 [13] Curtis Villamizar, "OSPF Optimized Multipath (OSPF-OMP)," Internet Draft draft-ietf-ospf-omp-02.txt, Feb., 1999.
 [14] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, A. Sastry, "The COPS(Common Open Policy Service) Protocol," Internet Draft draft-ietf-rap-cops-07.txt, Aug., 1999.

최 미 라

e-mail : mira01@ewha.ac.kr
 2001년 이화여자대학교 컴퓨터학과 졸업 (학사)
 2002년~현재 이화여자대학교 과학기술 대학원 컴퓨터학과(공학석사)
 관심분야 : MPLS 트래픽 엔지니어링, QoS routing, 혼잡제어 메커니즘





김 성 하

e-mail : sungha77.kim@samsung.com

2000년 이화여자대학교 컴퓨터학과 졸업
(학사)

2002년 이화여자대학교 과학기술대학원
컴퓨터학과(공학석사)

2002년~현재 삼성전자 정보통신 총괄 근무

관심분야 : QoS routing, cdma system, flow control



이 미 정

e-mail : lmj@mm.ewha.ac.kr

1987년 이화여자대학교 전자계산학과
졸업(학사)

1989년 University of North Carolina at
Chapel Hill 컴퓨터학과(석사)

1994년 North Carolina State University
컴퓨터공학과(공학박사)

1994년~현재 이화여자대학교 공과대학 컴퓨터학과 부교수

관심분야 : 고속 통신 프로토콜 설계 및 성능 분석, 비디오 전송
을 위한 트래픽 제어, 인터넷에서의 QoS 지원, Ad-
hoc 네트워크