

확장성 있는 동적 QoS 지원 프로토콜 (A Scalable Dynamic QoS Support Protocol)

문새롬^{*} 이미정^{**}

(Saerom Moon) (Meejeong Lee)

요약 인터넷에서 멀티미디어 응용들이 증가함에 따라 서비스 품질(QoS: Quality of Service)을 보장해 줄 수 있는 여러 가지 방안이 제안되어 왔는데, 일반적으로 이들 방안은 확장성과 종단간 QoS 보장 간에 상호 모순이 발생하는 경향을 가지고 있다. 이에, 본 논문에서는 확장성 있는 QoS 지원 프로토콜로서 DQSP(Dynamic QoS Support Protocol)를 제안한다. DQSP는 코어 라우터에서는 각 소스 에지 라우터 별 자원 할당 상태 정보만을 유지하도록 하고, 소스 에지 라우터가 각 링크들에 대하여 자신에게 할당된 자원의 양을 알고 이에 대해 동적인 사용 정보 유지하면서 이를 이용하여 연결 수락 결정을 내리도록 한다. 따라서, DQSP에서는 종단간 QoS를 보장하기 위한 자원 예약과 연결 수락 제어가 네트워크 코어에서의 시그널링을 수반하지 않고 네트워크 에지에서만 이루어진다. 즉, DQSP는 QoS 보장을 위해 사용자 플로우 별 입장 제어와 자원 할당을 지원하면서도 시그널링과 상태 정보 저장에 관계된 코어의 복잡성이 동적으로 변화하는 사용자 플로우 수에 따라 증가하는 것을 피할 수 있다는 면에서 확장성을 제공한다. 시뮬레이션을 통하여 DQSP와 기존의 자원 할당 및 공유 방식들을 비교한 결과 DQSP가 시그널링과 상태 정보 유지를 위한 오버헤드 면에서 기존의 종단간 QoS 보장을 제공하는 구조인 IntServ에서의 같은 사용자 플로우 수 증가에 대한 확장성 문제를 발생시키지 않으면서도 효율적으로 네트워크 자원을 활용할 수 있음을 확인할 수 있었다.

키워드 : 서비스 품질, 연결 수락 제어, 자원 예약, 동적 서비스 품질 지원, 서비스 품질 보장

Abstract As the number of multimedia applications increases, various protocols and architectures have been proposed to provide QoS(Quality of Service) guarantees in the Internet. Most of these techniques, though, bear inherent contradiction between the scalability and the capability of providing QoS guarantees. In this paper, we propose a protocol, named DQSP(Dynamic QoS Support Protocol), which provides the dynamic resource allocation and admission control for QoS guarantees in a scalable way. In DQSP, the core routers only maintain the per source-edge router resource allocation state information. Each of the source-edge routers maintains the usage information for the resources allocated to itself on each of the network links. Based on this information, the source edge routers perform admission control for the incoming flows. For the resource reservation and admission control, DQSP does not incur per flow signaling at the core network, and the amount of state information at the core routers depends on the scale of the topology instead of the number of user flows. Simulation results show that DQSP achieves efficient resource utilization without incurring the number of user flow related scalability problems as with IntServ, which is one of the representative architectures providing end-to-end QoS guarantees.

Key words : QoS, admission control, resource reservation, dynamic QoS support, QoS guarantee

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

^{*} 비회원 : 이화여자대학교 컴퓨터학과
992COG12@mm.ewha.ac.kr

^{**} 정회원 : 이화여자대학교 컴퓨터학과 교수
lmj@mm.ewha.ac.kr

논문접수 : 2002년 3월 18일

심사완료 : 2002년 5월 28일

1. 서론

오늘날의 인터넷은 최선형 서비스(Best Effort Service)라 불리는 단일 서비스만을 제공한다. 그러나 최선형 서비스로는 최근 새롭게 등장하고 있는 인터넷 전화나 멀티미디어 회의, 오디오나 비디오 스트리밍 등의 지연(delay)이나 지연 변이(delay variation)에 민감

한 응용들을 제대로 지원 할 수 없다. 이들 새로운 멀티 미디어 응용들은 일정한 종단간 전송 서비스의 품질(QoS: Quality of Service)을 보장받을 수 있어야 하는데, 이를 위해 인터넷에서 최선형 서비스와는 차별화된 QoS 서비스를 제공할 수 있는 방안이 필요하게 되었다. 이에 본 논문에서는 플로우별 홉-바이-홉 시그널링 없이 하나의 도메인 내에서 연결 수락 제어와 자원 예약을 수행하여 종단간 QoS를 보장할 수 있는 방안을 제안하고 이를 DQSP(Dynamic QoS Support Protocol)라 이름하였다. DQSP는 종단간 QoS 보장을 위한 사용자 플로우별 입장 제어와 자원 예약을 제공하면서도 코어 라우터의 복잡성이 사용자 플로우 개수가 아닌 네트워크 규모, 즉 에지 라우터 수에 의해 결정된다는 측면에서 확장성이 있다. DQSP는 링크 자원을 각 에지 라우터 별로 할당하고 에지 라우터가 자신에게 할당된 링크 자원에 관한 정보를 유지하도록 한다. 에지 라우터는 네트워크 상에서 자신이 사용할 수 있는 자원의 양과 사용하고 있는 자원에 관한 정보를 유지하면서 이 정보에 기반하여 사용자가 요청한 목적지로의 경로 상에 자신에게 할당된 자원 가운데 새로운 사용자 플로우를 받아들여도 될 만큼 충분한 자원이 남아 있다면 사용자 요청에 대한 연결을 수락한다. 즉, DQSP는 가용 자원에 대한 정보를 유지하고 이에 의해 연결 수락을 제어하므로 사용자의 QoS 보장이 가능할 뿐 아니라 사용자 플로우 별 시그널링 없이 에지 라우터에서 매 사용자 플로우별로 직접적으로 연결 수락 결정을 내리기 때문에 연결 설정 시간이 짧고 홉-바이-홉 시그널링 오버 헤드도 없다. 또한 DQSP에서는 효율적인 네트워크 자원 활용을 위하여 에지 라우터가 트래픽 부하에 따라 자신에게 할당되어 있는 자원을 반납하거나 추가적인 자원을 제공받을 수 있도록 한다. 만약 국부적인 네트워크 혼잡으로 인해 이와 같은 에지 라우터 간의 자원 재분배만으로는 사용자 요청을 더 이상 수용할 수 없는 경우에는 DQSP는 경로 재계산을 수행하여 현재의 라우팅에 의해서는 활용이 상대적으로 낮은 부분을 활용할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어서 2장에서는 기존의 연구에서 인터넷에서의 QoS 제공을 위해 제안되어 온 방안들을 소개한다. 3장에서는 DQSP를 적용하기 위한 네트워크 구조 및 링크 자원의 구성과 네트워크의 각 엔티티에서 유지해야 하는 정보를 소개하고 상세한 동작 방식을 설명한다. 4장에서는 DQSP의 성능 평가를 위한 시뮬레이션 모델을 제시하고 시뮬레이션 수행 결과를 분석한다. 마지막으로 5장에서는 결론과 함께 앞으로의 연구 방향을 기술한다.

2. 관련 연구

현재 인터넷에서 QoS를 제공하기 위한 방안은 크게 IntServ(Integrated Services)처럼 종단간 마이크로 플로우별 자원 예약 프로토콜을 사용하는 방법과 DiffServ(Differentiated Services)와 같이 정적으로 트래픽 클래스 별 자원을 예약하는 방법, IntServ와 DiffServ를 혼합하여 쓰는 방법, RSVP(Resource ReSerVation Protocol)를 집계된(aggregated) 트래픽에 적용하여 동적으로 트래픽을 예약하는 방법 등으로 나누어 볼 수 있다[1]. 마이크로 플로우별 자원 예약 프로토콜의 가장 대표적인 예로는 IntServ[2, 3, 4]에서 사용하는 RSVP[5, 6, 7]가 있는데, RSVP의 시그널링은 홉-바이-홉으로 플로우의 전송 경로를 경유하며 이 전송 경로 상의 각 라우터가 개별적인 트래픽 플로우별로 소프트 스테이트의 자원 예약 상태 정보를 유지해야 한다. 따라서 RSVP는 라우터의 기능 요구 사항이 높을 뿐 아니라 사용자 플로우 수에 관련한 확장성에 문제가 있다. 이와 같은 IntServ와 RSVP 모델의 확장성 문제를 완화하기 위해 RSVP를 마이크로 사용자 플로우가 아닌 집계된 플로우에 적용하는 방안이 제시되었다[8]. 그러나 이와 같은 방안에서는 적정 트래픽 집결의 단위를 결정하는 것이 문제이고, 집계된 플로우에 대한 동적인 자원 예약 및 연결 수락제어는 여전히 RSVP와 같은 홉-바이-홉 시그널링 오버헤드를 유발하며 코어 라우터들도 여전히 그 정도의 차이는 있으나 사용자 플로우 수에 비례하는 양의 상태 정보를 유지해야 한다.

DiffServ[9]는 근본적으로 IntServ와는 완전히 다른 접근 방식으로서 개별적인 플로우 별로 QoS를 제공한다는 개념에서 벗어나 네트워크 내에서는 트래픽 클래스(class)별로 서비스를 차별화하는 기능만을 제공한다. DiffServ에서는 트래픽 조절의 복잡한 기능은 망의 경계에서만 수행하고 네트워크의 내부에서는 아주 간단한 패킷 전달 기능만을 수행한다. 따라서, DiffServ에서는 IntServ의 확장성 문제가 완전히 해소되었지만, 네트워크가 단지 클래스별로 서비스를 차별화 할 뿐 종단간 QoS를 보장하지 못한다는 한계점이 있다[10].

IntServ와 DiffServ가 혼합된 환경에서는 DiffServ 망이 IntServ를 이용해 종단간 QoS를 보장하기 위한 네트워크 도메인의 하나로 간주되는데, DiffServ는 어떠한 표준화된 연결 수락 제어 방안도 제공하지 않고 인터넷의 혼잡 제어 문제에 대한 대책을 제시하지 않기 때문에 에지 노드는 DiffServ 도메인에 속한 노드의 트래픽 부하를 알 수 없다는 문제점이 있다[11, 12]. 이 문제를 해결하기 위한 방안으로 GRIP(Gauge&Gate Reservation

with Independent Probing)[13], 대역폭 브로커(Bandwidth Broker, BB)[14], RMD(Resource Management in DiffServ)[15] 등이 제안되었다. 먼저 GRIP에서는 소스 노드가 데이터 패킷을 보내기 전에 목적지로의 탐색 패킷을 네트워크에 내보내어 목적지에 이르는 경로 상태를 파악한다. 목적지는 탐색 패킷을 받으면 소스 노드에게 피드백 패킷을 보내는데, 만일 소스가 일정 시간동안 탐색 패킷에 대한 피드백 패킷을 수신하지 못하면 네트워크에 혼잡이 있다고 생각하고 연결 요청을 거절한다. GRIP은 DiffServ 도메인에 대한 연결 수락 제어 방법이 될 수는 있으나 자원 예약 방안은 제시하지 못한다. GRIP과는 달리 대역폭 브로커(Bandwidth Broker, BB)는 DiffServ 망에 대한 자원 예약과 연결 수락 제어를 모두 제공하기 위해 제안되었다. 이 방법은 각 도메인 내에 자원 관리자인 대역폭 브로커를 두고 이 대역폭 브로커가 도메인 내의 자원을 관리하고 이웃 도메인과 서비스 협약(Service Level Agreement, SLA)을 담당하도록 한다. 대역폭 브로커를 사용할 경우 연결 수락 제어는 자원 예약 메커니즘이 어떤 것이냐에 따라 결정되는데 실제 도메인 안에서 어떤 식으로 자원 예약을 할 것인가는 관리자의 자율에 맡기고 있다. 대역폭 브로커를 사용하는 방법은 DiffServ 네트워크에서 자원 예약과 연결 수락 제어를 위한 기본적인 접근 방식을 제공하지만 구체적인 자원 예약이나 연결 수락 제어를 위한 메커니즘은 아직 정의된 바 없다. RMD는 DiffServ 망에서 자원 예약과 연결 수락 제어를 위해 제안된 또 다른 방법이다. RMD는 진입 에지 라우터에서 진출 에지 라우터로 PHB(Per Hop Behavior)별로 자원을 예약하기 위한 PHR(Per Hop Reservation) 프로토콜과 PDR(Per Domain Reservation) 프로토콜로 구성된다. 진입 에지 라우터가 진출 에지 라우터로 자원 요청 메시지를 보내면 중간 라우터들은 자원이 있으면 자원을 예약하고 자원이 없으면 패킷에 표시를 한다. 이를 받은 진출 에지 라우터는 패킷에 표시된 정보를 살펴보고 진입 에지 라우터에게 연결 수락 결정을 알린다. 이 방안은 DiffServ 도메인에서는 네트워크 코어에서 전혀 사용자 플로우에 대한 상태 정보를 유지할 필요가 없어 상태 정보 유지 측면에서는 확장성이 있으나, 매 사용자 플로우에 대하여 홉-바이-홉 시그널링 프로토콜이 필요하므로 연결 수락 제어 시간이 길며 네트워크 코어에서의 시그널링 오버헤드는 사용자 플로우 수에 비례하여 커지게 된다.

3. DQSP(Dynamic QoS Support Protocol)

이 장에서는 사용자 플로우 수에 비례하는 네트워크

코어의 복잡성 증가 없이 사용자 플로우 별 입장 제어 및 자원 예약을 지원하는 DQSP의 상세한 구조와 동작에 대하여 설명한다. 3.1절에서 먼저 DQSP를 적용하기 위한 네트워크 구성과 링크 자원의 계층적인 구성 및 DQSP의 기본적인 동작 방식에 대하여 설명하고, 3.2절에서 DQSP 네트워크의 각 엔터티에서 유지하는 데이터 구조에 관하여 설명한다. 3.3절에서는 DQSP의 세 가지 기본 동작에 관하여 상세히 설명하며, 마지막으로 3.4절에서는 DQSP의 확장성 및 QoS 지원 정도를 IntServ 및 DiffServ와 비교하여 설명한다.

3.1 DQSP를 위한 네트워크 구성과 링크 자원의 구성

그림 1은 제안하는 DQSP를 적용하기 위한 네트워크 구성을 보인 것이다. 네트워크는 코어 라우터와 네트워크의 경계에 위치한 에지 라우터로 구성되고, 사용자는 에지 라우터를 통해서 네트워크에 접근한다. 코어 네트워크에는 링크 자원 할당을 담당하는 정책 서버를 두는데, 정책 서버는 경로를 계산하고, 네트워크의 모든 링크들에 대하여 각각 그 링크의 자원을 어떻게 할당할 것인지를 결정한다. 또한, 라우팅과 자원 할당에 관한 정보를 이를 필요로 하는 각 에지 라우터와 코어 라우터에게 알린다.

에지 라우터는 각 링크에 대하여 자신에게 할당된 자원의 양과 그 가운데 얼마를 사용하였는지에 관한 정보를 유지한다. 새로운 사용자 요청이 발생하면 에지 라우터는 자신이 유지하고 있는 라우팅 정보와 자원 사용에 관한 정보를 이용하여 그 사용자 요청이 사용해야 될 경로 상에 현재 충분한 가용 자원이 남아 있는지를 점검해 연결 수락 결정을 내린다. 에지 라우터는 자신에게 할당된 자원이 지나치게 많은 경우에는 이를 정책 서버에게 반납하고, 반대로 부족한 경우에는 정책 서버에게 추가 할당을 요청한다. 정책 서버는 이와 같은 에지 라우터의 자원 재구성 요청을 처리하여 링크 자원 할당을 재구성한다. 또한, 링크의 자원 할당에 변화가 있을 때

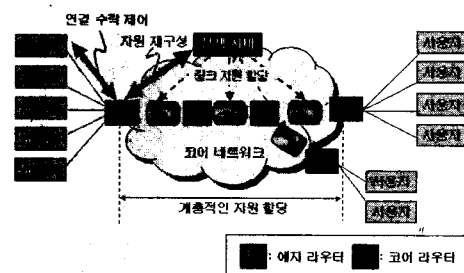


그림 1 DQSP의 네트워크 구성

마다 이를 해당 코어 라우터들과 에지 라우터들에게 알린다. 그리고, 필요한 경우 현재 자원 할당 상태를 기반으로 경로 계산을 다시 함으로써, 네트워크 자원 활용의 효율성과 사용자 요구 수용율의 향상을 도모한다.

DQSP에서는 정책 서버가 관리하는 네트워크의 링크 자원을 그림 2와 같이 계층적으로 구성한다고 가정한다. 먼저 각 물리적 링크는 논리적으로 몇 개의 QoS 클래스로 분리되고, 각 QoS 클래스에 대해서는 DiffServ와 같이 통계적인 부하 예측에 따라 장기적으로 고정된 양의 네트워크 자원이 설비된다. 각 QoS 클래스 내에서는 다시 ISP들 별로 특정 양의 링크 자원을 설비한다고 가정한다. 또한, 각 QoS 클래스 내에서 하나의 사용자 플로우를 위해 할당하는 대역폭의 양은 일정하다고 가정한다. 이와 같은 가정은 실제적으로 적용될 가능성이 높은 네트워크 자원 설비 방식에 기반한 것이다. 각 ISP가 특정 클래스를 위해 사용하는 링크 자원을 논리적으로 프로비전드 링크라 부르기로 한다. DQSP의 정책 서버는 각 프로비전드 링크의 자원을 다시 그 프로비전드 링크를 사용하는 에지 라우터들에게 각각 트렁크라는 논리적 썩션으로 나누어 할당하고, 이들 트렁크는 동적으로 재구성한다. 즉, DQSP에서는 네트워크의 자원을 계층적으로 3 단계-물리적 링크, 프로비전드 링크, 트렁크-로 분류하고, ISP 별 QoS 클래스에 할당되는 자원인 프로비전드 링크까지는 장기적으로 고정적인 양의 자원이 할당된다고 가정하며, 네트워크의 활용율을 높이고 사용자의 QoS 지원을 향상시키기 위해 프로비전드 링크 내에서 동적으로 트렁크를 재구성한다.

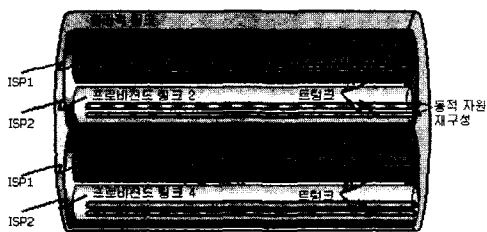


그림 2 DQSP에서의 링크 자원의 계층적인 구성

그림 3은 DQSP에서 트렁크를 이용해 사용자 플로우들이 소스 에지 라우터로부터 목적지 에지 라우터로 전달되는 예를 보여주고 있다. 그림 3의 네트워크는 3개의 에지 라우터와 3개의 코어 라우터 그리고 6개의 논리적인 프로비전드 링크들로 구성된다. 에지 라우터 쌍들을 각각 연결하는 굵은 실선들은 현재 정책 서버에 의해

결정되어 있는 각 소스 에지 라우터로부터 목적지 에지 라우터로의 경로들인데, 경로는 소스 에지 라우터로부터 목적지 에지 라우터에 이르기 위해 사용하는 트렁크들의 연결로 구성된다. DQSP의 정책 서버는 하나의 소스와 목적지 라우터 쌍 간에 여러 개의 경로를 사용하도록 할 수 있는데, P_{ij}^k 는 소스 에지 라우터 E_i 에서 목적지 에지 라우터 E_j 로의 k 번째 경로를 뜻한다. 라우팅이 결정되면 이에 따라 각 프로비전드 링크를 지나는 경로들이 결정되는데, DQSP는 각 프로비전드 링크에 대하여 그 프로비전드 링크를 경유하는 경로들의 소스 에지 라우터들에게 프로비전드 링크의 자원을 나누어 할당한다. 예를 들어 프로비전드 링크 PL_1 에는 소스 에지 라우터 E_1, E_2, E_3 으로부터의 경로들이 지나간다. 이와 같은 경우, DQSP는 프로비전드 링크 PL_1 의 자원을 세 개의 트렁크 T_{11}, T_{12}, T_{13} 으로 나누고 각각 소스 에지 라우터 E_1, E_2, E_3 을 위해 할당한다. T_{ij} 는 프로비전드 링크 PL_i 에서 에지 라우터 E_j 에 할당된 트렁크를 뜻한다. 각 소스 에지 라우터는 새로운 사용자 플로우가 발생할 때마다 그 사용자 플로우에 대하여 FID(Flow Identification)를 할당한다. FID_{ij}^k 는 경로 P_{ij}^k 를 경유하는 1번째 사용자 플로우를 뜻한다. 그림 3에서는 E_1 에서 E_3 으로의 경로 P_{13}^1 상에 하나의 사용자 플로우 FID_{13}^1 이 존재한다.

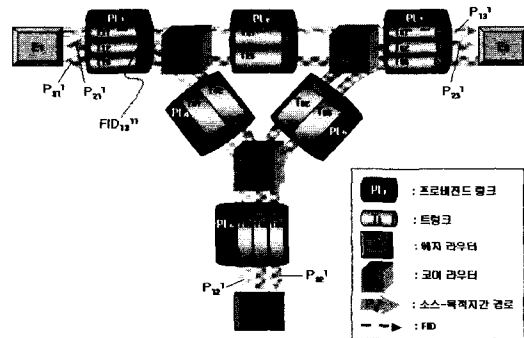


그림 3 DQSP 네트워크에서 링크 자원 사용의 예

3.2 DQSP의 각 엔티티가 유지하는 정보

DQSP는 정책 서버에 자원 할당 테이블과 경로 구성 테이블을 유지하고 에지 라우터에는 자원 사용 테이블과 경로 구성 테이블을, 코어 라우터에는 자원 할당 테이블과 라우팅 테이블을 각각 유지한다. 정책 서버의 자원 할당 테이블에는 네트워크의 모든 프로비전드 링크에 대하여 각각 현재 그 프로비전드 링크의 자원이 어

스 에지 라우터에게 얼마만큼 할당되어있는지의 정보를 기록하는데, 그림 4는 그림 3과 같은 네트워크의 경우 초기화된 자원 할당 테이블의 예를 보여준 것이다. 자원 할당 테이블의 프로비전드 링크 필드에는 프로비전드 링크의 ID(Identification)와 프로비전드 링크에 장기적으로 할당된 고정적인 용량, 그리고 프로비전드 링크의 용량 가운데 현재 각 트렁크에 할당해 준 자원 양의 합을 각각 기록한다. 또한, 각 프로비전드 링크에 대해서는 프로비전드 링크를 경유해 사용자 플로우를 라우트하는 소스 에지 라우터들의 ID 리스트(공유 에지 라우터 필드)와 해당 트렁크들에 관한 정보(트렁크 필드)를 기록한다. 예를 들어, 그림 3의 프로비전드 링크 PL₂의 용량이 300이라면 자원 할당 테이블에서 해당 프로비전드 링크의 용량 필드에는 300이 기록되고, PL₂에 현재 소스 에지 라우터 E₁과 E₃으로부터의 경로가 존재하므로 이들의 ID가 공유 에지 라우터 필드에 기록된다. 또한, 정책 서버는 초기화시 각 프로비전드 링크의 용량의 일부를 그 프로비전드 링크를 공유하는 트렁크 간에 동일하게 분배한다. 따라서 만일 초기에 프로비전드 링크 용량의 100%를 모두 트렁크에 할당한다면, PL₂의 공유 에지 라우터 E₁과 E₃을 위한 트렁크 T₂₁, T₂₃에 각각 150 씩의 자원이 할당되어 PL₂의 트렁크 필드에 그림 4와 같이 기록된다. 그리고, 현재 프로비전드 링크 PL₂에서 트렁크에 할당된 총 자원 양이 300이므로 프로비전드 링크 필드의 '할당된 자원'에는 300이 기록된다. 트렁크 필드의 '연결 라우터'에는 각 트렁크가 연결하는 두 인접 라우터를 기록한다. 추후 트렁크 구성이 동적으로 변화해 나감에 따라 프로비전드 링크 필드의 할당된 자원 값과 트렁크 필드의 할당된 자원의 값이 변화해 나가게 된다.

ISP	정책적 클래스	프로비전드 링크			트렁크				
		ID	용량	할당된 자원	공유 에지 라우터	ID	연결 라우터	할당된 자원	
1	Q ₁	PL ₁	300	300	E ₁	T ₁₁	E ₁	C ₁	100
					E ₂	T ₁₂	C ₁	E ₁	100
					E ₃	T ₁₃	C ₁	E ₁	100
		PL ₂	300	300	E ₁	T ₂₁	C ₁	C ₃	150
					E ₃	T ₂₃	C ₃	C ₁	150
					E ₁	T ₃₁	C ₃	E ₃	100
		PL ₃	300	300	E ₂	T ₃₂	C ₃	E ₃	100
					E ₃	T ₃₃	E ₃	C ₃	100
					E ₁	T ₄₁	C ₁	C ₂	150
		PL ₄	300	300	E ₂	T ₄₂	C ₂	C ₁	150
					E ₂	T ₅₂	C ₂	C ₃	150
					E ₃	T ₅₃	C ₃	C ₂	150
		PL ₅	300	300	E ₁	T ₆₁	C ₂	E ₂	100
					E ₂	T ₆₂	E ₂	C ₂	100
					E ₃	T ₆₃	C ₂	E ₂	100

그림 4 정책 서버의 자원 할당 테이블의 예

정책 서버의 경로 구성 테이블에는 모든 소스/목적지 쌍에 대하여 경로 정보를 유지한다. 즉, 각 소스와 목적지 에지 라우터에 대하여 현재 계산되어 있는 모든 경로들에 대한 경로 ID 그리고 각 경로에 대하여 그 경로가 경유하는 코어 라우터들과 경로를 구성하는 프로비전드 링크 ID의 리스트를 기록한다. 그림 5는 그림 3의 네트워크에 대한 경로 구성 테이블 예를 보인 것이다. 동적으로 경로가 재계산됨에 따라 경로 구성 테이블에는 새로운 경로 정보가 추가되고, 이에 따라 자원 할당 테이블의 공유 에지 라우터 필드 및 트렁크 필드도 변화해 나간다.

ISP	정책적 클래스	소스	목적지	경로					
				ID	경유 코어 라우터	프로비전드 링크			
1	Q ₁	E ₁	E ₂	P ₁₂ ¹	C ₁	C ₂	PL ₁	PL ₄	PL ₆
				P ₁₃ ¹	C ₁	C ₃	PL ₁	PL ₂	PL ₃
				P ₂₁ ¹	C ₂	C ₁	PL ₆	PL ₄	PL ₁
				P ₂₃ ¹	C ₂	C ₃	PL ₆	PL ₅	PL ₃
				P ₃₁ ¹	C ₃	C ₁	PL ₃	PL ₂	PL ₁
				P ₃₂ ¹	C ₃	C ₂	PL ₃	PL ₅	PL ₆

그림 5 정책 서버의 경로 구성 테이블의 예

에지 라우터는 경로 결정과 사용자 요청의 연결 수락 제어율 위해 경로 구성 테이블과 자원 사용 테이블을 유지한다. 그림 6과 7은 각각 그림 3의 에지 라우터 E₁의 경로 구성 테이블과 자원 사용 테이블을 예로 보인 것이다. 경로 구성 테이블에는 모든 목적지들에 대해 경로 ID를 기록하고 각 경로에 대해서는 다시 그 경로가 어떠한 트렁크들로 구성되어 있는지와 현재 그 경로를 이용하고 있는 플로우들의 FID, 그리고 각 플로우들이 사용하고 있는 자원은 얼마인지에 대한 정보를 유지한다. 자원 사용 테이블에는 해당 에지 라우터에게 할당되어 있는 모든 트렁크에 대하여 할당된 자원의 양과 사용중인 자원의 양을 기록한다. 에지 라우터는 정책 서버가 경로 재계산을 수행하여 새로운 경로를 알려 오면 새로운 경로에 관한 정보를 경로 구성 테이블에 추가하고 트렁크 별 자원 할당 정보를 자원 사용 테이블에 저장한다. 또한, 사용자 요청이 수락되어 새로운 FID가 생성되면 이 정보를 경로 구성 테이블의 FID필드에 추가하고, 해당 플로우가 경유하는 각 트렁크에 대하여 자원 사용 테이블의 사용중인 자원 값을 증가시킨다. 반대로, 사용자 플로우가 종료되면 경로 구성 테이블에서 해당 FID의 엔트리를 삭제하고, 자원 사용 테이블에서는 그 플로우가 사용한 모든 트렁크에 대하여 각각 사용중인 자원 값을 감소시킨다.

ISP	트래픽 클래스	목적지	경로				FID	
			ID	구성 트렁크	ID	사용중인 자원		
1	Q ₁	E ₂	P ₁₂ ¹	T ₁₁	T ₄₁	T ₆₁	-	-
		E ₃	P ₁₃ ¹	T ₁₁	T ₂₁	T ₃₁	FID ₁₃ ¹¹	5

그림 6 에지 라우터의 경로 구성 테이블의 예

ISP	트래픽 클래스	트렁크		
		ID	할당된 자원	사용중인 자원
1	Q ₁	T ₁₁	100	5
		T ₂₁	150	5
		T ₃₁	100	5
		T ₄₁	150	-
		T ₆₁	100	-

그림 7 에지 라우터의 자원 사용 테이블의 예

코어 네트워크에서 정책 서버가 결정한 경로를 따라 정책 서버가 할당한 자원을 이용하여 사용자 플로우가 전달되도록 하기 위하여 코어 라우터에는 라우팅 테이블과 자원 할당 테이블을 유지한다. 코어 라우터는 라우팅 테이블의 정보를 이용하여 사용자 플로우를 라우트하고, 각 트렁크 별로 정책 서버가 할당한 자원에 따라 사용자 플로우에 대한 서비스를 제공한다. 그림 8과 9는 각각 그림 3의 코어 라우터 C_i이 유지하는 라우팅 테이블과 자원 할당 테이블을 보여준 것이다. 라우팅 테이블에는 각 경로 ID에 대해 그 경로를 사용하는 사용자 플로우를 어느 프로비전드 링크의 어느 트렁크로 라우트시켜야할지의 정보를 유지하고, 자원 할당 테이블에는 각 프로비전드 링크에 대하여 트렁크 별로 할당된 자원의 양을 기록한다. 정책 서버가 동적으로 경로를 재계산하거나 트렁크를 재구성함에 따라 코어 라우터의 라우팅 테이블과 자원 할당 테이블은 변화하게 된다.

경로 ID	프로비전드 링크	트렁크
P ₁₂ ¹	PL ₄	T ₄₁
P ₁₃ ¹	PL ₂	T ₂₁
P ₂₁ ¹	PL ₁	T ₁₂
P ₃₁ ¹	PL ₁	T ₁₃

그림 8 코어 라우터의 라우팅 테이블

ISP	트래픽 클래스	프로비전드 링크	트렁크	할당된 자원
1	Q ₁	PL ₁	T ₁₂	100
			T ₁₃	100
		PL ₂	T ₂₁	150
		PL ₄	T ₄₁	150

그림 9 코어 라우터의 자원 할당 테이블

3.3 DQSP의 동작

DQSP의 동작은 크게 경로 설정, FID 결정, 트렁크 재구성 등의 세 가지로 구성된다. 경로 설정은 정책 서버가 에지 라우터간의 경로를 계산하고 계산된 경로에 기반하여 링크의 트렁크 자원을 구성하며 경로와 자원 할당 정보를 라우터들에게 알리는 과정이다. 이 과정은 초기화 시에 일단 한 번 행해지고, 정책 서버가 에지 라우터의 트렁크 재구성 요청을 들어 줄 수 없을 경우 다시 수행한다. FID 결정 과정은 에지 라우터가 사용자 플로우의 QoS 요청에 적합한 전송 경로를 결정하는 과정으로 에지 라우터에서 새로운 사용자 플로우가 발생할 때마다 행해진다. 트렁크 재구성 과정은 에지 라우터에 의해 트리거 되며 이 과정에서 에지 라우터는 정책 서버에게 자원 활용률이 높은 트렁크를 위해서 해당 프로비전드 링크에서의 추가적인 자원을 요청하거나 자원 활용률이 낮은 트렁크의 자원을 반납한다. 다음에서는 위의 세 가지 과정에 대해 각각 자세히 설명한다.

3.3.1 경로 설정

그림 10은 경로 설정 과정을 요약한 것이다. 초기화시 정책 서버는 프로비전드 링크들로 구성된 논리적 네트워크 토폴로지에서부터 각 소스와 목적지 에지 라우터간의 최단 경로를 계산하여 경로 구성 테이블에 이 정보를 저장한다. 일단 경로가 결정되면 정책 서버는 각 프로비전드 링크에 대하여 그 프로비전드 링크를 점유하는 모든 경로들을 알 수 있고 또한 이들 경로의 소스 에지 라우터들이 어떤 것들인지를 알 수 있다. 임의의 프로비전드 링크에 대하여 그 프로비전드 링크를 점유하는 경로들의 소스 에지 라우터들을 그 프로비전드 링크에 대한 '공유 에지 라우터'라 부르기로 한다. 초기화시 정책 서버는 각 프로비전드 링크의 자원의 일부를 그 프로비전드 링크에 대한 공유 에지 라우터들 간에 동일하게 분배하고 각 에지 라우터 별로 트렁크를 정의하는데, 이와 같은 정보들

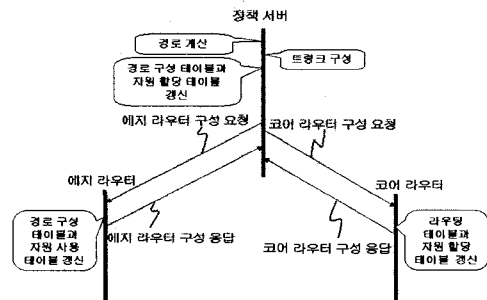


그림 10 경로 설정 과정

메시지 형태	에지 라우터 구성 요청
에지 라우터 ID	E1
QoS 클래스	Q1
경로 정보	
목적지	E2
경로 ID	P12'
구성 트링크	T11, T41, T61
목적지	E3
경로 ID	P13'
구성 트링크	T11, T21, T31
자원 할당 정보	
트링크 ID	T11
할당된 자원	100
트링크 ID	T21
할당된 자원	150
트링크 ID	T31
할당된 자원	100
트링크 ID	T41
할당된 자원	150
트링크 ID	T61
할당된 자원	100

그림 11 에지 라우터 구성 요청 메시지의 예

은 정책 서버의 자원 할당 테이블에 기록된다. 경로 혹은 트링크에 대한 자원 할당에 변화가 있는 경우 정책 서버는 이를 해당 에지 라우터들과 코어 라우터들에게 각각 에지 라우터 구성 요청 메시지와 코어 라우터 구성 요청 메시지를 보내어 알린다.

에지 라우터 구성 요청 메시지는 그림 11에서 보여준 것처럼 어느 소스 에지 라우터에 대한 구성 요청 메시지인지를 표시하기 위한 '에지 라우터 ID' 필드와 구성 대상이 된 QoS 클래스를 표시하기 위한 'QoS 클래스' 필드, 그리고 해당 소스-에지 라우터로부터 각 목적지에 이르는 경로 정보를 표시하기 위한 '경로정보' 필드 및 해당 에지 라우터에 할당된 모든 트링크들에 대해 할당된 자원 양을 표시하기 위한 '자원 할당 정보' 필드 등으로 이루어진다. 정책서버는 자신이 유지하는 경로 구성 테이블에서 우선 에지 라우터 구성 요청 메시지를 보내고자 하는 목적지 에지 라우터에 대한 경로 정보를 알아내는데, 경로 구성 테이블에서 (그림 5 참조) 발견할 수 있는 경로 정보는 프로비전드 링크의 리스트로 이루어진다. 정책서버는 경로를 구성하는 이들 프로비전드 링크들을 각각 다시 자원 할당 테이블에서 (그림 4 참조) 찾아 각 프로비전드 링크에서 해당 소스 에지 라우터를 위해 할당되어 있는 트링크와 자원 양을 알아내어 경로 정보 필드에는 경로 ID와 경로를 구성하고 있는 트링크의 리스트를 표시하고, 자원 할당 정보 필드에는 트링크 별로 할당된 자원 양을 기록한다. 그림 11은 정책 서버가 소스 에지 라우터 E1에게 보내는 예

메시지 형태	코어 라우터 구성 요청
코어 라우터 ID	C1
QoS 클래스	Q1
경로 정보	
경로 ID	P12'
프로비전드 링크	PL4
트링크	T41
경로 ID	P13'
프로비전드 링크	PL2
트링크	T21
경로 ID	P21'
프로비전드 링크	PL1
트링크	T12
경로 ID	P31'
프로비전드 링크	PL1
트링크	T13
자원 할당 정보	
트링크 ID	T12
할당된 자원	100
트링크 ID	T13
할당된 자원	100
트링크 ID	T21
할당된 자원	150
트링크 ID	T41
할당된 자원	150

그림 12 코어 라우터 구성 요청 메시지의 예

지 라우터 구성 요청 메시지의 예이다.

코어 라우터 구성 요청 메시지는 그림 12에서 보여준 것과 같이 어느 코어 라우터에 대한 구성 요청 메시지인지를 표시하기 위한 '코어 라우터 ID' 필드와 구성 대상이 된 QoS 클래스를 표시하는 'QoS 클래스' 필드, 그리고 해당 코어 라우터를 경유하는 모든 경로들에 대해 그 코어 라우터를 경로 상의 다음 홉에 연결하는 프로비전드 링크와 트링크 ID를 표시해 주는 '경로 정보' 필드 및 이들 트링크 각각에 대한 자원 할당 정보를 기록하는 '자원 할당 정보' 필드 등으로 구성된다. 정책 서버는 코어 라우터 구성 요청 메시지를 준비하기 위하여 각 코어 라우터에 대하여 경로 구성 테이블에서 (그림 5 참조) 그 코어 라우터를 경유하는 모든 경로에 대하여 경로 상의 다음 홉에 이르기 위한 프로비전드 링크들을 알아내고, 이들 프로비전드 링크들을 각각 자원 할당 테이블에서 (그림 4 참조) 찾는다. 그리고 이들 각 프로비전드 링크에 대한 자원 할당 테이블 엔트리에서 '연결 라우터' 필드를 보고 그 프로비전드 링크 내에서 해당 코어 라우터가 사용하게될 모든 트링크 ID 및 각 트링크에 할당된 자원 양을 알아내어 코어 라우터 구성 요청 메시지를 작성한다. 그림 12는 정책 서버가 C1에게 보내는 코어 라우터 구성 요청 메시지를 예로 보인

것이다. 에지 라우터 구성 요청 메시지를 받은 에지 라우터는 이를 이용하여 자신의 경로 구성 테이블과 자원 사용 테이블을 변경하고, 코어 라우터 구성 요청 메시지를 받은 코어 라우터는 이를 이용해 자신의 라우팅 테이블과 자원 할당 테이블을 갱신한다.

에지 라우터는 자신에게 할당되어진 어떤 트렁크의 자원 활용률이 임계치보다 높으면 그 트렁크에 추가 자원을 제공받기 위해 정책 서버에게 트렁크 재구성 요청을 보내는데(트렁크 재구성에 관해서는 3.3.3에서 설명함), 만일 재구성이 요청된 트렁크가 속한 프로비전드 링크에 남아 있는 자원(그림 4의 자원 할당 테이블에서 프로비전드 링크 필드의 '용량' - '할당된 자원')이 트렁크 재구성 요청을 들어주기에 충분하지 않다면 정책 서버는 경로 재계산을 수행한다. 정책 서버는 자원 할당률이 임계치 이상인 프로비전드 링크를 제외시킨 논리적 네트워크에 대하여 추가 자원이 요청된 트렁크를 지나는 모든 경로들에 대해 경로 재계산을 수행해서 새로 결정된 경로들을 경로 구성 테이블에 추가하고, 새로운 경로가 점유하는 모든 프로비전드 링크에서 추가 자원 할당을 요청한 소스 에지 라우터에 대응하는 트렁크에 할당되는 자원의 양을 요청된 자원의 양만큼 늘림으로써 트렁크를 재구성한다. 추가 자원이 요청된 트렁크를 지나는 경로들을 찾기 위해 정책 서버는 먼저 자원 할당 테이블로부터 해당 트렁크가 속한 프로비전드 링크를 알아내고, 다시 경로 구성 테이블에서 자원 재구성을 요청한 소스 에지 라우터를 소스로 하며 해당 프로비전드 링크를 점유하는 경로들을 찾아낸다. 예를 들어 그림 3에서 에지 라우터 E₁이 정책 서버에게 트렁크 T₁₁에 대한 자원 추가를 요청하였는데 T₁₁이 속한 프로비전드 링크 PL₁에 남아 있는 자원 양이 그 요청을 들어 주기에 부족하다면, 정책 서버는 먼저 T₁₁이 속한 프로비전드 링크 PL₁을 자원 할당 테이블로부터 알아내고 다시 경로 구성 테이블로부터 E₁을 소스로 하고 PL₁을 점유하는 모든 경로들(P₁₂¹과 P₁₃¹)을 파악하여 이들 경로의 소스와 목적지 쌍(E₁과 E₂, E₁과 E₃)에 대하여 경로 재계산을 수행한다. 재계산된 경로 상의 모든 프로비전드 링크에서는 에지 라우터 E₁에 대응하는 트렁크에 할당되는 자원 양을 원래 E₁이 T₁₁에 대하여 자원 추가를 요청했던 양만큼 증가시킨다.

3.3.2 FID 결정

그림 13은 FID 결정 과정을 요약한 것이다. 사용자 플로우가 발생하면 사용자 노드는 소스 에지 라우터에게 경로 요청 메시지를 보내어 목적지에 이르기 위한 경로를 요청한다.

경로 요청 메시지는 그림 14에서 보여 주듯이 사용자

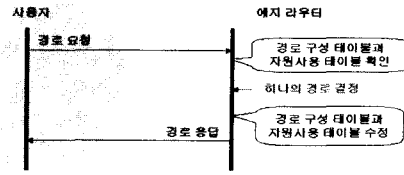


그림 13 FID 결정 과정

를 식별하기 위한 '사용자 ID' 필드와 목적지를 표시하기 위한 '목적지 에지 라우터' 필드, 그리고 QoS 클래스를 표시하기 위한 'QoS 클래스' 필드 및 몇 번째 요청인지를 표시하는 '요청 회수' 필드로 구성된다. 그림 14는 사용자가 에지 라우터 E₁에게 E₃으로의 Q₁ 클래스를 위한 경로를 요청할 경우의 경로 요청 메시지의 예이다. 경로 요청 메시지를 받은 에지 라우터는 경로 요청 메시지의 목적지 에지 라우터 필드를 보고 자신이 유지하는 경로 구성 테이블과 자원 사용 테이블에서 요청된 목적지로의 경로 중 사용자가 요구하는 QoS 클래스를 위한 충분한 대역폭이 존재하는 경로가 있는지 확인해 사용자 요청을 수락한다. 사용자 요청을 수락한 에지 라우터는 자신의 자원 사용 테이블에서 해당 경로를 구성하는 트렁크들에 대하여 '사용중인 자원'을 증가시키고, 경로 구성 테이블의 해당 경로 엔트리에 새로이 결정된 FID에 대한 정보를 추가한다. 에지 라우터는 사용자에게 경로 응답 메시지를 보내어 사용자의 경로 요청에 대한 수락 여부를 알리고, 사용자 경로 요청이 수락되었을 경우에는 결정된 경로 ID와 FID도 알린다. 그림 15는 그림 3의 예에서 그림 14의 경로 요청이 수락되어 P₁₃¹에 FID₁₃¹²가 생성되었을 경우, 에지 라우터 E₁이 사용자에게 보내는 경로 응답 메시지이다.

메시지 형태	경로 요청
사용자 ID	사용자 5
목적지 에지 라우터	E ₃
QoS 클래스	Q ₁
요청 회수	1

그림 14 경로 요청 메시지의 예

메시지 형태	경로 응답
사용자 ID	사용자 5
목적지 에지 라우터	E ₃
QoS 클래스	Q ₁
요청 수락 여부	수락
경로 ID	P ₁₃ ¹
FID	FID ₁₃ ¹²

그림 15 경로 응답 메시지의 예

경로 응답 메시지를 받은 사용자는 경로 응답 메시지의 요청 수락 여부 필드를 보고 자신의 경로 요청이 수락되었는지 확인한다. 사용자는 자신의 경로 요청이 거절되었으면 재시도 간격 후 해당 플로우를 위한 경로 요청 메시지를 다시 에지 라우터에게 보내는데, 이와 같은 재시도는 일정 최대 회수까지 이루어지게 된다. 사용자는 에지 라우터로부터 경로 요청이 수락되었음을 통보 받으면 자신의 요청에 대하여 결정된 경로 ID와 FID 필드를 확인하고, 결정된 경로 ID와 결정된 FID를 이용해 데이터 전송을 시작한다. 에지 라우터는 사용자 데이터를 받으면 자신의 경로 구성 테이블을 보고 사용이 허가된 FID인지 확인하고, 사용이 허가된 FID라면 사용자가 요청한 경로상의 다음 홉으로 데이터를 전송한다. 코어 라우터는 경로 ID를 이용해 라우팅 테이블에서 다음 노드로의 프로비전드 링크와 트렁크를 확인하고 전송한다. 데이터 전송이 끝나면 사용자는 소스 에지 라우터에게 그림 16과 같은 FID 해제 요청 메시지를 보낸다. FID 해제 요청 메시지를 받은 에지 라우터는 자신의 경로 구성 테이블에서 해제가 요청된 FID의 모든 정보를 삭제하고, 자원 사용 테이블에서 해당 경로를 구성하는 트렁크들에 대하여 '사용중인 자원' 값을 감소시킨다.

메시지 형태	FID 해제 요청
사용자 ID	사용자 5
경로 ID	P13 ¹
FID	FID13 ¹²

그림 16 FID 해제 요청 메시지의 예

3.3.3 트렁크 재구성

그림 17은 트렁크 재구성 과정을 요약한 것이다. 트렁크 재구성은 에지 라우터에 의해 트리거된다. 에지 라우

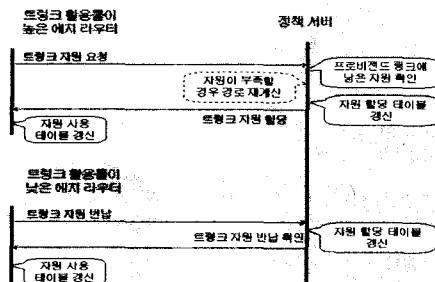


그림 17 트렁크 재구성 과정

터는 자신에게 할당된 어떤 트렁크의 자원 활용률이 특정 임계치 이상이 되면 해당 트렁크에 자원을 추가로 제공받기 위해 정책 서버에게 트렁크 자원 요청 메시지를 보내고, 반대로 자원 활용률이 특정 임계치 이하가 되는 트렁크가 발생하면 정책 서버에게 해당 트렁크 자원의 일부를 반납한다고 알리는데, 이와 같은 에지 라우터의 요청에 의하여 정책 서버는 트렁크를 재구성하게 된다.

정책 서버는 에지 라우터로부터 그림 18과 같은 트렁크 자원 요청 메시지를 받으면 에지 라우터 ID와 트렁크 ID 필드를 이용해 자신의 자원 할당 테이블을 검색해서 해당 프로비전드 링크를 찾아낸다. 해당 프로비전드 링크에 남아 있는 자원이 추가 자원을 필요로 하는 트렁크가 속한 QoS 클래스의 한 사용자 플로우를 위한 대역폭 양의 2배 이상이면 정책 서버는 에지 라우터의 트렁크 자원 요청을 수락하고, 해당 트렁크의 QoS 클래스에 속하는 하나의 사용자 플로우의 대역폭 요구량의 2배만큼의 자원을 추가로 할당한다. 만일 프로비전드 링크에 남아 있는 자원이 충분하지 않다면, 에지 라우터의 트렁크 자원 요청은 거절되고, 경로 재계산이 트리거된다.

정책 서버는 에지 라우터의 트렁크 자원 요청에 대하여 그림 19와 같은 트렁크 자원 할당 메시지를 보낸다. 예를 들어, 정책 서버가 에지 라우터 E₁로부터 그림 18과 같은 트렁크 자원 요청 메시지를 받으면, 정책 서버는 자신이 유지하는 그림 4의 자원 할당 테이블에서 에지 라우터 E₁의 트렁크 T₁₁이 속해 있는 프로비전드 링크 PL₁의 용량과 현재 이미 할당된 자원 양을 확인한다. 만약 T₁₁과 함께 PL₁을 공유하고 있는 T₁₂와 T₁₃에 트래픽 부하가 낮아 T₁₂와 T₁₃이 자원을 반납해서 현재 PL₁에 남아 있는 자원의 양이 트렁크 T₁₁이 속한 QoS 클래스의 하나의 사용자 플로우가 요구하는 양의 두 배 이상이라면 정책 서버는 T₁₁에 추가 자원을 할당하고 E₁에게 그림 19와 같은 트렁크 자원 할당 메시지를 보낸다.

에지 라우터의 트렁크 자원 요청이 수락되면 정책 서

메시지 형태	트렁크 자원 요청
에지 라우터 ID	E ₁
트렁크 ID	T ₁₁

그림 18 트렁크 자원 요청 메시지의 예

메시지 형태	트렁크 자원 할당
에지 라우터 ID	E ₁
트렁크 ID	T ₁₁
요청 수락 여부	수락

그림 19 트렁크 자원 할당 메시지의 예

버는 자신의 자원 할당 테이블에서 해당 트렁크에 대한 '할당된 자원' 필드와 해당 트렁크가 속한 프로비전드 링크의 '할당된 자원' 필드를 갱신한다. 또한, 정책 서버로부터 트렁크 자원 할당 메시지를 받은 에지 라우터는 자신의 트렁크 자원 요청이 수락되었는지 확인하고, 만일 트렁크 자원 요청이 수락되었다면 자신의 자원 사용 테이블에서 해당 트렁크에 대한 '할당된 자원' 필드를 변경한다.

한편 에지 라우터는 FID 해체에 의하여 어떤 트렁크의 활용율이 임계치 이하가 된 것을 발견하면 그림 20과 같은 트렁크 자원 반납 메시지를 정책 서버에게 보낸다. 이를 받은 정책 서버는 자원 할당 테이블에서 자원 반납이 요청된 트렁크에 해당하는 '할당된 자원' 필드와 그 트렁크가 속한 프로비전드 링크의 '할당된 자원' 필드를 갱신한다. 예를 들어 정책 서버가 E₁로부터 그림 50의 트렁크 자원 반납 메시지를 받았다면, 그림 4의 자원 할당 테이블에서 T₂₁의 '할당된 자원'을 100으로 조정하고, T₂₁이 속한 프로비전드 링크 PL₂의 '할당된 자원'을 250으로 변경한다. 정책 서버는 자원 할당 테이블을 갱신한 후 트렁크 자원 반납 메시지를 보낸 에지 라우터에게 트렁크 자원이 반납되었음을 알리기 위해 그림 21과 같은 트렁크 자원 반납 확인 메시지를 보낸다. 이를 받은 에지 라우터는 자신의 자원 사용 테이블에서 자원 반납이 확인된 트렁크의 '할당된 자원' 필드의 값을 50만큼 감소시킨다.

메시지 형태	트렁크 자원 반납
에지 라우터 ID	E ₁
트렁크 ID	T ₂₁
반납하는 자원	50

그림 20 트렁크 자원 반납 메시지의 예

메시지 형태	트렁크 자원 반납 확인
에지 라우터 ID	E ₁
트렁크 ID	T ₂₁
반납된 자원	50

그림 21 트렁크 자원 반납 확인 메시지의 예

3.4 DQSP의 확장성 및 QoS 지원 정도

DQSP는 모든 사용자 플로우들이 걸집되는 네트워크의 코어 부분에 있는 라우터에서는 그림 8과 9에서 볼 수 있듯이 마이크로 플로우별 정보가 아닌 에지 라우터별 자원 할당 정보만을 유지한다. 각 개별 사용자 플로우에 대한 정보는 네트워크의 에지에 있는 라우터들만이 각자 자신에게 직접 연결되어 있는 사용자 플로우들에 한해서만 정보를 유지하면 된다. 즉, 네트워크 코어에서의 복잡성이 사용자 플로우가 아닌 그보다 훨씬 그 수가 적고 덜 동적인 에지 라우터 수에 의해 결정된다. 따라서 DQSP는 코어 네트워크에서 유지해야 하는 상태 정보 면에서 IntServ의 확장성 문제를 피할 수 있다.

또한, DiffServ가 네트워크 상의 모든 에지 라우터들로부터 유입되는 모든 플로우들의 집합에 대해 클래스별로 통계적 예상 트래픽 부하에 근거해 반 고정적으로 자원을 설비해 두는 것에 반해, DQSP에서는 코어 라우터의 자원이 각 에지 라우터별로 할당되기 때문에 임의의 코어 라우터에서 자원을 공유하는 플로우 집합은 한 에지 라우터에서 발생한 동 클래스 트래픽 가운데 그 코어 라우터를 경유해 라우트되는 플로우들로 국한된다. 또한, 실제적인 자원 사용이 어떻게 이루어지는가에 따라 동적으로 에지 라우터간에 자원을 재분배할 뿐 아니라, 경로 재계산을 발생하여 현재의 라우팅에 의해서는 활용이 상대적으로 낮은 부분을 활용할 수 있도록 라우팅을 조정해 나가기 때문에 좀 더 효율적인 네트워크 자원 활용이 가능하다. 그리고, 에지 라우터가 자원 사용 정보를 유지하고 이를 이용하여 트래픽 플로우가 경유해야 하는 경로 상에 가용 자원이 남아 있는 경우에만 연결을 수락하므로 코어 네트워크에서의 시그널링 없이 연결 수락과 자원 예약이 이루어진다.

그리고, 표 1은 DQSP와 IntServ, DiffServ를 종합적으로 비교한 것이다.

4. 시뮬레이션 모델 및 결과 분석

DQSP의 자원 활용의 효율성 및 시그널링 오버헤드

표 1 DQSP, IntServ, DiffServ의 비교

	QoS 지원을 위한 코어 라우터의 복잡성을 결정하는 요인	예약된 자원을 공유하는 플로우 집결 단위	연결 수락 제어 및 자원 할당
DQSP	에지 라우터 수, QoS 클래스 수	하나의 에지 라우터에서 발생하는 동 QoS 클래스에 속하는 플로우들의 집결	직접적 (각 플로우에 대하여 가용 자원이 있는 경우에만 연결 수락)
IntServ	사용자 플로우 수	하나의 사용자 플로우	직접적 (각 플로우에 대하여 가용 자원이 있는 경우에만 연결 수락)
DiffServ	QoS 클래스 수	모든 에지 라우터에서 발생하는 동 QoS 클래스에 속하는 플로우들의 집결	통계적

를 파악하는 것을 목표로 시뮬레이션을 수행하였다. 시뮬레이션은 OPNET을 이용하여 구현되었으며, Windows 2000 시스템에서 수행하였다. DQSP와의 비교를 위하여 CS(Complete Sharing)와 CP(Complete Partitioning) 등의 두 가지의 극단적인 자원 활용 모델을 구현하였는데, CS는 기존의 IntServ의 자원 활용 모델에 해당하는 모델이라 할 수 있고 CP는 DiffServ의 자원 활용 모델과 유사하게 정적으로 자원을 할당하는 모델이다. 4.1절에서 먼저 시뮬레이션 모델을 설명하고, 4.2절에서는 시뮬레이션 결과를 분석한다.

4.1 시뮬레이션 모델

DQSP와의 자원 활용 효율성 비교를 목적으로 구현된 두 가지의 모델 가운데 기존의 IntServ의 자원 활용 모델에 해당하는 CS는 각 QoS 클래스를 위해 미리 자원을 할당해 두지 않고 모든 QoS 클래스가 네트워크의 가용 자원을 공유하도록 하며, 자원 예약은 RSVP 휴-바이-휴 신호 프로토콜을 통해 플로우별로 이루어지도록 한다. 반면, CP는 DiffServ와 유사하게 네트워크 자원을 미리 할당하여 이를 장기적으로 유지하는 모델이다. 그러나, DiffServ의 경우 QoS 클래스 별로 자원을 할당하는 것과는 달리 본 시뮬레이션에서 구현한 CP의 자원 할당 단위는 DQSP와 동일한 트렁크이다. 단, DQSP는 동적으로 트렁크를 재구성하는데 반해 CP에서는 트렁크 구성을 장기적으로 고정시켜 둔다. 따라서, CP에서는 초기 트렁크 구성이 성능에 큰 영향을 미치는데, 이 점을 감안하여 본 시뮬레이션에서는 트렁크 별 평균 트래픽 부하에 비례하도록 트렁크 자원을 구성하는 최적의 트렁크 구성을 가지는 경우와 DQSP의 초기 트렁크 구성과 동일하게 트렁크를 구성한 경우의 두 가지 CP 모델을 실험해 보았다.

CS 모델에서 사용하는 RSVP의 리프레쉬 간격 및 RSVP 상태 정보의 라이프 타임은 [7]에서 제시한 디플트 값 권고에 따랐고, 사용자 플로우를 위한 경로 계산은 QoS를 위한 OSPF 확장(QOSPF) 방안에서의 벨만포드 알고리즘[16]을 사용하였다. QOSPF의 네트워크 상태 정보 교환과 라우트 재계산에 관련된 여러 가지 파라미터 값은 [17]의 권고 사항과 [18]을 참고하였다. 즉, 이웃에게 라우터가 자신의 존재를 알리는 HELLO 메시지의 전달 간격은 20초로 설정하였고, 네트워크 상태 정보를 알리는 LSA(Link State Advertisement) 메시지는 RSVP에 의해 링크에 예약된 자원 양이 60% 이상 변할 경우 플러딩하도록 하였다. 또한 타이머를 이용하여 LSA가 적어도 100초마다 한 번씩은 이웃 노드에게 보내지도록 하고, 각 라우터는 최소한 100초 마다 한 번씩은 경로를 재계산하되 새로운 네트워크 상태 변

화를 발견할 때마다 경로를 재계산하도록 하였다[18]. CP 모델에서는 미리 각 에지 라우터에서 모든 목적지 라우터에 대하여 최단 경로를 한 번 계산해 두고 모든 사용자 요청에 대하여 이 경로 정보를 사용하도록 하였다. DQSP에서는 매 100초마다 트렁크의 자원 활용률을 검사하도록 하였고, 만일 트렁크의 자원 활용률이 1/3 이하이면 할당된 트렁크 자원의 1/3을 반납하도록 하였다. 또한, 트렁크에 남아 있는 자원이 해당 QoS 클래스 하나의 플로우에 대한 대역폭 요구에 미치지 못할 경우 트렁크에 추가 자원을 요청하도록 하였다.

그림 22는 시뮬레이션을 위해 사용된 네트워크 토폴로지이다. 코어 네트워크는 네 개의 라우터들과 이들을 연결하는 링크, 그리고 정책 서버로 구성되며, 네 개의 에지 라우터들이 사용자 그룹 엔터티들을 네트워크에 연결하고 있다. 각 라우터에서의 제어 패킷 별 프로세싱 시간은 0.00512초로 설정하였고 라우터 간의 전파 지연 시간은 0.0033초라 가정하였다. 네트워크에는 하나의 ISP만이 존재한다고 가정하고, ISP의 비디오 트래픽을 위한 프로비전드 링크에는 72M 비트의 대역폭을 할당하고, 오디오 트래픽을 위한 프로비전드 링크에는 2.56M 비트의 대역폭을 할당한다고 가정하였다. 각 사용자 그룹 엔터티에서는 비디오와 오디오 두 가지 트래픽 클래스에 속하는 세션을 발생시키는데 이를 이 후부터 '사용자 요청'이라 부르기로 한다. 비디오 트래픽과 오디오 트래픽 세션에 대한 사용자 요청은 1:5의 비율로 발생한다고 가정하였고, 이 두 가지의 사용자 요청 전체에 대한 사용자 요청 발생 간격은 고정 값인 경우와 다양한 평균 값을 가지는 지수 분포의 랜덤 변수인 경우 모두를 실험하였다. 비디오 세션의 대역폭 요구는 3M 비트로 설정하였고 세션 지속 시간은 평균이 1152초인 지수 분포 랜덤 변수라 가정하였다. 오디오 세션의 대역폭 요구는 64K 비트로 설정하였고 세션 지속 시간

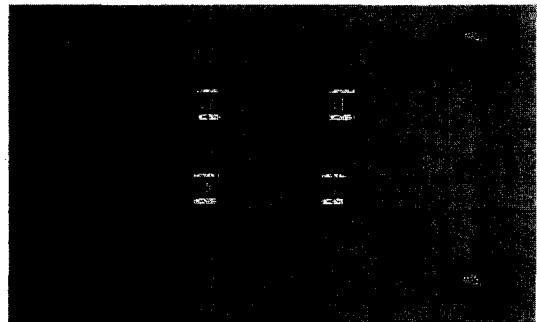


그림 22 네트워크 토폴로지

은 평균이 300초인 지수 분포 랜덤 변수라 가정하였다.

위와 같은 가정 하에서 각 에지 라우터에서 발생하는 트래픽 부하를 다양하게 변화시켜보며 사용자 요청 거절률과 연결 설정 시간, 그리고 제어 메시지 오버헤드를 측정하였다. 먼저, 요청 거절률은 전체 사용자 요청에 대한 거절된 사용자 요청 수의 비율로서 측정하였고, 연결 설정 시간은 사용자가 요청을 보낸 시간부터 이 요청에 대한 최종 연결 수락 결정이 이루어지기까지 소요되는 시간으로 측정하였다. DQSP에서 에지 라우터는 사용자 플로우를 위한 적당한 경로를 발견하지 못하면 사용자에게 요청이 거절되었음을 알리는데, 요청 거절을 보고 받은 사용자는 재시도 시간 간격 후에 다시 에지 라우터에게 요청을 보낸다. 만일 최대 재시도 회수만큼 시도를 했는데도 에지 라우터가 사용자 요청에 대한 적당한 경로를 발견하지 못하면 최종적으로 사용자 요청은 거절되었다고 판단한다. 이와 같은 경우의 사용자 플로우의 연결 설정 시간은 최초 요청 시간으로부터 최종 거절을 표시하는 경로 응답 메시지를 수신하기까지 걸린 시간으로 한다. 본 시뮬레이션에서는 DQSP의 최대 재시도 회수를 3으로 설정하였으며, 재시도 간격은 평균이 15초인 지수 분포 랜덤 변수로 두었다. CS 모델에서는 연결 수락 제어를 위해 RSVP 신호 프로토콜을 사용하므로 사용자 플로우의 연결 설정 시간은 그 사용자 플로우를 위한 PATH 메시지가 홈-바이-홈으로 목적지까지 전달되고, 이 PATH 메시지를 받은 목적지로부터 소스까지 RESV 메시지가 다시 전달되기까지 소요된 시간이며, 만일 연결 수락을 위한 최대 대기 시간 동안 소스에서 RESV 메시지를 받지 못하면 사용자의 요청이 거절되었다고 판단되고, 이 경우 연결 설정 시간은 최대 연결 수락 대기 시간이 된다. CS 모델에서는 연결 수락 결정을 기다리는 최대 사용자 대기 시간을 60초로 설정하였는데, 이는 DQSP에서 사용자 요청이 거절되는 경우의 평균 연결 설정 대기 시간에 해당하며, 시뮬레이션에서 사용한 네트워크 모델에서 RSVP의 홈-바이-홈 신호 프로토콜에 의하여 연결 수락 결정이 이루어지기에 충분한 시간이다. CP 모델의 연결 수락 제어는 DQSP에서와 같은 방식으로 에지 라우터에서 이루어진다.

제어 메시지 오버헤드로는 시뮬레이션 시간 전체에 걸쳐 네트워크 각 엔터티에서 전달한 모든 제어 메시지의 수를 측정하였다. DQSP 모델에서는 3.3절에서 설명된 모든 제어 메시지들의 수를 측정하였고, CS 모델에서는 RSVP 시그널링을 위한 메시지들과 QOSPF 라우팅을 위한 메시지들의 수를 측정하였다. CP 모델의 경우에는 제어 메시지 오버헤드가 없다.

4.2 시뮬레이션 결과

이 장에서는 시뮬레이션의 결과를 DQSP의 자원 재구성 효과를 보여주는 것과, CS 및 CP 자원 할당 모델과의 성능 비교를 보여주는 것으로 구분하여 설명한다.

4.2.1 DQSP의 자원 재구성 효과

이 실험에서는 한 개의 에지 라우터만을 활성 에지 라우터로 두고 나머지 에지 라우터들은 비활성 에지 라우터로 만들어 시간이 흐름에 따라 활성 에지 라우터와 비활성 에지 라우터들에게 할당된 트렁크들에 대한 자원 할당과 활용률이 각각 어떻게 변해가는지를 살펴보았다. 활성 에지 라우터는 부하가 높은 에지 라우터로서 평균이 10인 지수 분포를 따르는 랜덤 변수에 의한 시간 간격마다 사용자 요청이 발생되도록 하였고, 비활성 에지 라우터는 부하가 상대적으로 매우 낮은 에지 라우터로서 매 300초마다 사용자 요청이 발생되도록 하였다. 트래픽 부하에 따른 DQSP의 효과를 보기 위해 6000초까지 사용자 요청이 발생되도록 하고 8000초까지 실험해 보았다. 따라서 네트워크 상의 트래픽 부하는 시뮬레이션 초기에 증가하다가 일정한 수준을 유지하며, 6000초 이후에는 감소하게 된다.

그림 22와 23은 라우터 1과 라우터 2사이에서 비디오 트래픽을 위해 할당된 프로비전드 링크의 트렁크 별 자원 활용률과 트렁크에 할당된 자원량의 변화를 보여준다. 이들 그림을 통하여 트렁크 활용률에 따라 트렁크에 할당된 자원의 양이 동적으로 변해 나감을 볼 수 있다. 즉, 활용률이 높은 트렁크에 할당되어진 자원은 점차 증가하며, 활용률이 낮은 트렁크에 할당되어진 자원은 점차 감소함을 볼 수 있다. 비활성 에지 라우터는 자신에게 할당된 트렁크의 자원 활용률이 1/3 이하이므로, 트렁크에 할당된 자원이 트렁크 당 최소 자원 할당량인 하나의 비디오 플로우의 대역폭 요구에 해당하는 3M 비트가 될까지 매 100초마다 할당되어있는 자원을 1/3 씩 줄여나간다. 반면에 활성 에지 라우터를 위한 트렁크는 남아 있는 자원이 단일 비디오 플로우 요구를 만족 시키기에 부족한 경우가 빈번하므로 정책 서버로부터 추가 자원을 할당받아 트렁크에 할당된 자원의 양이 50Mbps 이상이 된다. 그림 22와 23을 종합하여 볼 때, DQSP가 동적인 트렁크 재구성을 통하여 효율적으로 네트워크 자원을 활용함을 볼 수 있다.

4.2.2 DQSP와 CS 및 CP 자원 할당 모델의 비교

DQSP와 CS 및 CP를 비교하기 위한 시뮬레이션에서는 모든 에지 라우터에 비슷한 정도의 사용자 요청이 발생하는 경우와 어떤 한 에지 라우터에서 다른 에지 라우터들 보다 많은 양의 사용자 요청이 발생하는 경우

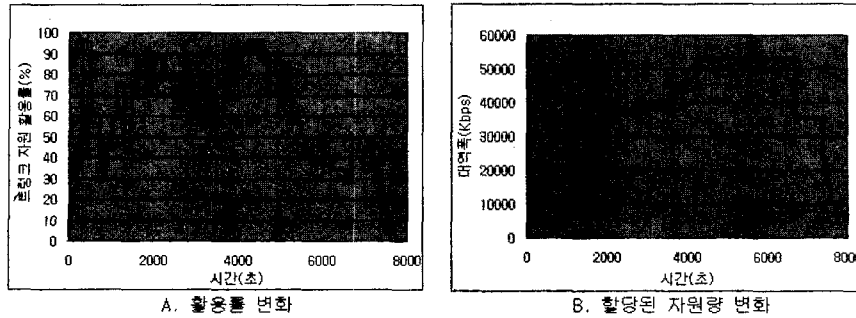


그림 23 활성 에지 라우터 트렁크의 활용률 변화와 할당된 자원량 변화

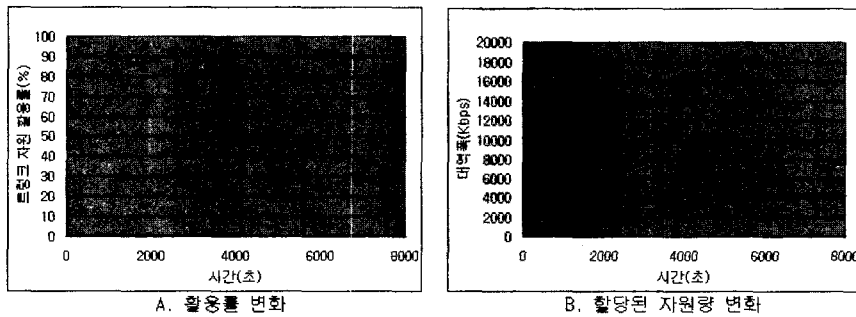


그림 24 비활성 에지 라우터 트렁크의 활용률 변화와 할당된 자원량 변화

의 두 가지 경우를 실험하였다. 두 경우 모두 사용자 요청 발생 간격은 지수 분포를 따른다. 사용자 요청이 고르게 발생하는 경우에 대한 시뮬레이션에서는 트래픽 부하 변화에 따른 성능 변화를 보기 위해 사용자 그룹 노드의 플로우 생성 간격의 평균을 20초 ~ 200초까지 변화시키며 각각 30000초씩 실험해 보았다. 특정 에지 라우터에서 발생하는 사용자 요청 양이 많은 경우에 대한 시뮬레이션에서는 부하가 높은 에지 라우터에 연결되어 있는 사용자 그룹의 노드에서는 평균 플로우 생성 간격을 20초로 설정하고, 나머지 사용자 그룹의 노드에서는 평균 플로우 생성 간격을 20초 ~ 200초로 변경시켜보며 각각 60000초씩 실험해 보았다.

그림 25와 26은 각각 사용자 요청 발생량이 에지 라우터간에 같은 경우와 차이가 있는 경우에 대하여 트래픽 부하 변화에 따른 사용자 요청 거절률의 변화를 보여준다. 두 경우 모두 완전히 동적인 방식으로 네트워크의 자원을 활용하는 CS 모델이 가장 낮은 사용자 요청 거절률을 보이는데, DQSP는 CS 모델과 거의 비슷하거나 약간 높은 사용자 요청 거절률을 보이고 두 가지의 CP보다는 낮은 사용자 요청 거절률을 보인다. CP 모델

의 경우에는 트래픽 부하가 높을 때 그 성능이 초기 트렁크 구성에 크게 영향을 받을 수 있다. 트래픽 부하가 에지 라우터들 간에 비슷한 경우에 대한 실험에서는 제안하는 DQSP의 성능과 최적의 구성을 사용하는

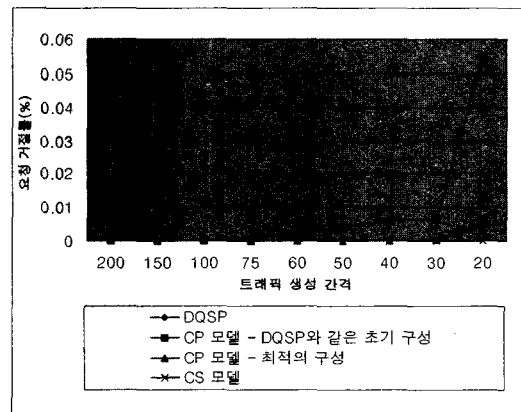


그림 25 에지 라우터들의 부하가 같은 경우의 트래픽 부하 변화에 따른 요청 거절률 변화

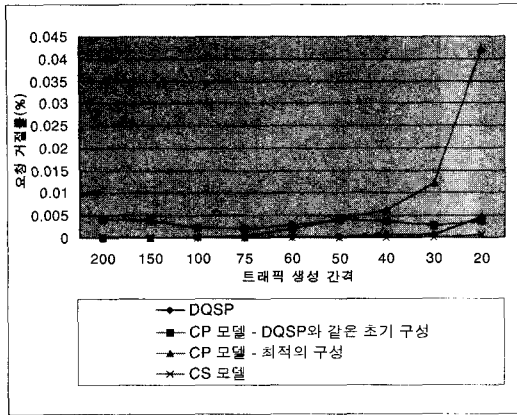


그림 26 한 에지 라우터의 부하가 상대적으로 높은 경우의 트래픽 부하 변화에 따른 요청 거절률 변화

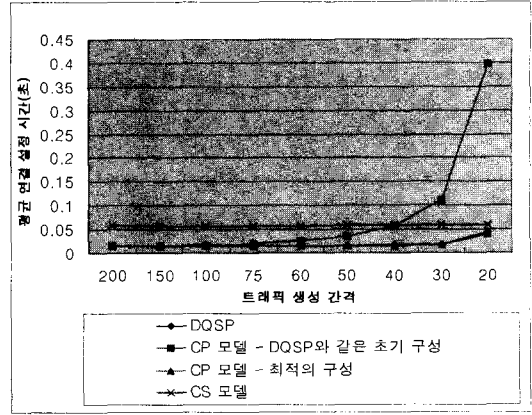


그림 27 에지 라우터들의 부하가 같은 경우의 트래픽 부하 변화에 따른 연결 설정 시간 변화

CP 모델의 성능에 차이가 미미하지만 트래픽 부하가 다른 경우의 실험에서는 DQSP가 확연히 낮은 사용자 요청 거절률을 보인다. 이는 CP 모델은 정적으로 자원을 할당하고 사용하지만, DQSP는 트렁크 수준에서 동적으로 자원을 재구성하여서 자원 활용이 효율적이기 때문이다. 그러나 트래픽 부하가 극도로 높아지는 경우에는 DQSP의 성능이 좋은 성능을 보이는 경우의 CP와 거의 일치함을 볼 수 있는데, 이는 트래픽 부하가 극도로 높을 경우에는 프로비전드 링크 자원이 모두 트렁크에 할당되어지고 트렁크의 자원 활용률 또한 높으므로 더 이상 트렁크 재구성이 이루어 질 수 없기 때문이다.

그림 27과 28은 각각 사용자 요청 발생량이 에지 라우터간에 같은 경우와 차이가 있는 경우에 대하여 트래픽 부하 변화에 따른 평균 연결 설정 시간의 변화를 보여준다. 본 시뮬레이션에서는 4.1절에서 정의한 연결 설정 시간 정의에 따라 거절된 요청에 대한 연결 설정 시간도 평균 연결 설정 시간 측정에 포함하였는데, 거절되는 사용자 요청의 경우 연결 설정 시간이 훨씬 길기 때문에 사용자 요청 거절률이 높아지면 거절된 사용자 요청에 대한 연결 설정 시간이 평균 연결 설정 시간에 크게 영향을 미친다. 따라서, 사용자 요청 거절률이 낮은 경우의 결과에서 연결이 수락된 사용자 요청의 경우 연결 설정 시간을 비교해 볼 수 있는데, 이 경우 CS 모델은 CP 모델이나 DQSP에 비해 2 ~ 3 배의 연결 설정 시간을 필요로 한다. 이는 CP 모델이나 DQSP의 경우에는 사용자 요청을 받는 에지 라우터가 자신이 유지하고 있는 자원 사용 정보에 기반 하여 홉-바이-홉 시그널링 없이 연결 수락 제어를 수행하는 반면, CS 모델은

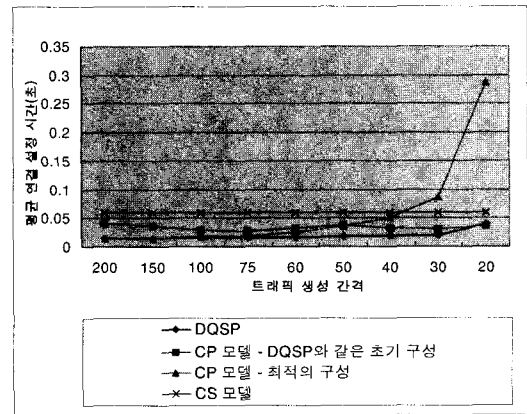


그림 28 한 에지 라우터의 부하가 상대적으로 높은 경우의 트래픽 부하 변화에 따른 연결 설정 시간 변화

연결 수락 제어를 위해 RSVP 홉-바이-홉 신호 프로토콜을 사용하기 때문이다. 즉, CS 모델에서는 연결 수락 제어를 위해 매 사용자 요청에 대하여 소스에서 목적지까지 PATH 메시지가 전달되고, 다시 목적지에서 소스까지 RESV 메시지가 전달되어야 한다. CP 모델의 경우 평균 세션 발생 간격이 50초 이하인 경우 즉 트래픽 부하가 높은 경우 평균 연결 설정 시간이 급격히 길어지고 DQSP의 경우도 약간 길어지게 되는데, 이는 거절되는 사용자 요청에 대한 연결 설정 시간의 영향 때문이다. 트래픽 부하가 높은 경우 CS는 사용자 요청 거절률이 상대적으로 낮기 때문에 거절된 사용자 요청에 대한 연결 설정 시간의 영향을 적게 받아 낮은 사용자 연결 설정 시간을 보인다.

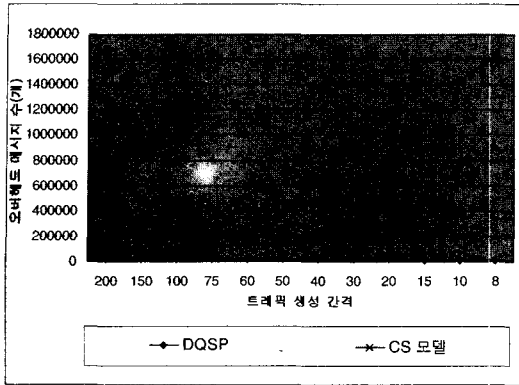


그림 29 에지 라우터들의 부하가 같은 경우의 트래픽 부하 변화에 따른 제어 메시지 오버헤드 변화

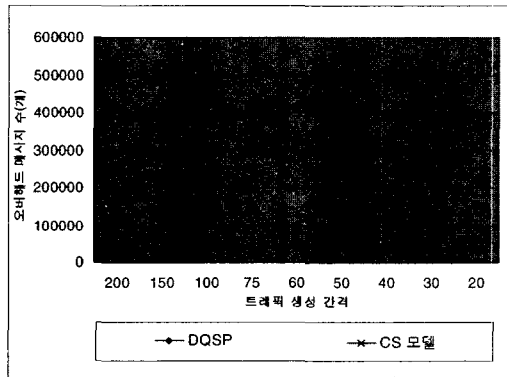


그림 30 한 에지 라우터의 부하가 상대적으로 높은 경우의 트래픽 부하 변화에 따른 제어 메시지 오버헤드 변화

그림 29와 30은 각각 사용자 요청 발생량이 에지 라우터간에 같은 경우와 차이가 있는 경우에 대하여 트래픽 부하 변화에 따른 제어 메시지 오버헤드의 변화를 보여준다. 두 경우 모두 CS의 오버헤드는 DQSP의 오버헤드의 5배 이상이 된다. 이는 CS의 경우 매 사용자 플로우에 대하여 RSVP 신호 프로토콜로 연결 수락 및 자원 할당을 수행하기 때문에 시그널링 오버헤드가 매 사용자 플로우에 대하여 발생하지만 DQSP의 경우에는 연결 수락을 위한 사용자 플로우 별 제어 메시지가 없기 때문이다. 또한, CS는 트래픽 부하가 증가함에 따라 오버헤드가 증가하는데, 이는 라우터에서 소프트 상태로 유지되는 PATH 상태 정보와 RESV 상태 정보를 유지하기 위해 보내지는 PATH 메시지, RESV 메시지 등의

RSVP 오버헤드가 플로우 수에 비례하여 증가하고, QOSPF를 위한 LSA 교환 오버헤드도 트래픽 부하 증가에 따라 증가하는 경향이 있기 때문이다. 이에 반하여 DQSP의 오버헤드는 트래픽 부하와 상관없이 일정한 수준을 유지하기 때문에 사용자 플로우 수에 대한 확장성이 있음을 볼 수 있다. 제어 메시지 오버헤드 뿐만 아니라, 유지해야 하는 상태 정보 면에서도 CS와 같은 사용자 플로우 별 자원 할당 모델에서는 코어 라우터가 사용자 플로우 별 정보를 유지해야 하므로 확장성 문제가 심각한 반면, DQSP에서는 에지 라우터 별 자원 할당 정보만을 유지하면 되므로 사용자 플로우 수에 대하여 확장성이 있다.

5. 결론 및 향후 연구 과제

본 논문에서는 종단간 QoS를 보장하기 위한 확장성 있는 방법으로서 DQSP를 제안하였다. DQSP는 한 도메인 내의 자원 예약과 연결 수락 제어 메커니즘을 정의한다. 제안하는 DQSP는 링크를 물리적 링크, 프로비전드 링크, 트렁크의 세 계층으로 나누고 각 ISP가 특정 클래스를 위해 사용하는 프로비전드 링크 수준까지는 정적으로 자원을 설비하는 것을 가정한다. 프로비전드 링크의 자원은 라우팅에 의하여 프로비전드 링크를 경유하는 경로를 가지는 소스 에지 라우터들 간에 공유하도록 하되, 소스 에지 라우터 별로 트렁크가 할당되고 트렁크에 대한 자원 할당은 동적으로 구성된다. 각 에지 라우터는 자신에게 할당된 트렁크들의 자원 사용 정보를 유지하면서 자신에게 도착하는 사용자 플로우에 대하여 자원 예약과 연결 수락 제어를 수행한다. 또한 자신에게 할당된 트렁크에 대하여 트래픽 부하에 따라 추가적으로 자원을 요청하거나 사용하지 않는 자원을 반납한다.

DQSP에서는 동적으로 트렁크를 재구성하기 때문에 네트워크 자원을 효율적으로 사용할 수 있으며, 각 에지 라우터가 자신을 위해 할당되어진 트렁크 자원에 관한 정보를 유지하면서 목적지로의 경로 상에 자원이 있을 경우에만 사용자 요청을 수락하기 때문에 사용자에게 종단간 QoS 보장을 제공할 수 있다. 또한 DQSP에서는 네트워크 코어에서 사용자 플로우가 아닌 에지 라우터 별 상태 정보만을 유지하고, 연결 수락 제어나 자원 예약을 위한 사용자 플로우 별 홉-바이-홉 시그널링을 사용하지 않기 때문에 사용자 플로우 수에 대해 확장성이 있다.

시뮬레이션을 통해 DQSP의 성능을 두 가지의 극단적인 자원 활용 모델인 CS 및 CP와 비교한 결과 DQSP의 자원 활용 효율성이 자원 공유 가능성을 극대화한 CS에 거의 근접함을 볼 수 있었다. 또한, CS 모

델에 비해 연결 설정 시간이 짧고 제어 메시지 오버헤드 현저히 작은 것을 볼 수 있었다. 특히, CS의 경우에는 트래픽 로드가 증가함에 따라 제어 메시지 오버헤드가 급격히 증가하는데 반하여 DQSP는 일정 수준을 유지하기 때문에 확장성이 좋음을 확인할 수 있었다.

DQSP는 단일 정책 서버에 의존하여 라우팅 및 트렁크 별 자원할당을 수행하므로, 네트워크 토폴로지의 규모에 대해서는 여전히 확장성 문제를 안고 있으며, 단일 정책 서버의 오류시에는 네트워크 전체 작동이 실패하게 된다는 문제가 있다. 향후 연구에서 정책 서버를 사용하지 않고 분산된 방법으로 DQSP를 수행하는 방안과 다중의 정책 서버를 두고 이들 간의 일치성을 유지하는 방안 등에 대하여 살펴보고자 한다. 또한, DQSP와 함께 사용될 수 있는 폴리싱, 버퍼링, 스케줄링 등을 설계·구현하고, 패킷 레벨의 시뮬레이션을 수행하여 DQSP의 사용자 QoS 보장 정도를 분석하고자한다.

참 고 문 헌

- [1] H. de Meer, G. Feher, N. Blefari-Melazzi, G. Karagiannis, D. Partian, L. Westberg, "Analysis of Existing QoS Solutions," Internet Draft (Work in Progress): draft-demeer-nsis-analysis-00.txt, 2001. 11.
- [2] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: An Overview," RFC1633, 1994. 6.
- [3] S. Shenker, C. Partidge and R. Guerin, "Specification of Guaranteed Quality of Service," RFC 2212, 1997. 12.
- [4] J. Wroclawski, "Specification of the Controlled-Load Network Element Service," RFC 2211, 1997. 12.
- [5] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol(RSVP) Version 1 Functional Specification," RFC 2205, 1997. 9.
- [6] J. Wroclawski, "The Use of RSVP with IETF Integrated Services," RFC 2210, 1997. 9.
- [7] Paul P. White, "RSVP and Integrated Services in the Internet: A Tutorial," IEEE Communications Magazine, 1997. 5.
- [8] F. Baker, C. Iturralde, F. le Faucheur, B. Davie, "Aggregation of RSVP for IPv6 and IPv4 Reservations," RFC 3175, 2001. 9.
- [9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services," RFC 2475, 1998, 12.
- [10] Walter Weiss, "QoS with Differentiated Service," Bell Labs Technical Journal, 1998.
- [11] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, E. Relstaine, "A Framework for Integrated Services Operation Over DiffServ Networks," RFC 2998, 2000. 11.
- [12] J. Wroclawski, A. Charney, "Integrated Service Mappings for Differentiated Services Networks," Internet Draft (Work in Progress): draft-ietf-issll-ds-map-01.txt, 2001. 2.
- [13] G. Bianchi, N. Blefari-Melazzi, "A Migration Path to provide End-to-End QoS over Stateless Networks by Means of a Probing-driven Admission Control," Internet Draft (Work in Progress): draft-bianchi-blefari-end-to-end-qos-02.txt, 2001. 11.
- [14] A. Terzis, L. Wang, J. Ogawa, L. Zhang, "A Two-Bit Resource Management Model for Internet," Global Internet 99, 1999. 12.
- [15] L. Westberg, M. Jacobsson, G. Karagiannis, S. Oosthoek, D. Partain, V. Rexhepi, R. Szabo, P. Wallentin, "Resource Management in Diffserv (RMD) Framework," Internet Draft (Work in Progress): draft-westberg-rmd-framework-00.txt, 2001. 4.
- [16] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda, T. Przygienda, "QoS Routing Mechanisms and OSPF Extensions," RFC 2676, 1999. 8.
- [17] J. Moy, "OSPF Version 2," RFC 2328, 1998. 4.
- [18] George Apostolopoulos, Sanjay Kamat, Roch Guerin, Satish K. Tripathi, "Quality of Service Based Routing: A Performance Perspective," SIGCOMM, 1998.



문 새 림

1999년 성신여자대학교 컴퓨터학과 졸업(이학사). 2002년 이화여자대학교 과학기술대학원 컴퓨터학과 졸업(공학석사) 2002년 ~ 현재 삼성전자 정보통신총괄 연구원. 관심분야는 IP QoS, 멀티미디어 통신

이 미 정

정보과학회논문지 : 정보통신
제 29 권 제 1 호 참조