

인터넷 차별화 서비스 네트워크에서의 효율적인 호 수락 제어 및 자원 이용 방안에 관한 연구

조병일[†], 유상조^{**}

요 약

DiffServ 네트워크는 기본적으로 명시적인 호 수락 제어 절차를 정의하고 있지 않다. 본 논문에서는 DiffServ 네트워크에서 플로우 기반의 호 수락 제어를 수행하되, 호 수락 제어 후에는 클래스 별 패킷 처리를 수행하는 새로운 메커니즘을 제안한다. 경로상의 라우터들은 클래스 별로 중첩되어 있는 트래픽과 호를 요구하는 신규 플로우의 트래픽을 중첩하여 도출한 소요 대역을 이용하여 호 수락 여부를 결정 하도록 하였으며 사용자의 다양한 QoS 요구 조건에 따른 소요 대역 계산 방법 및 파라미터 도출 방법을 사용하여 클래스의 우선 순위 에 따라 효율적으로 호 수락 제어를 할 수 있는 방안을 제안하였다. 또한 플로우 기반의 호 수락 제어 후에 DiffServ 본연의 클래스 별 패킷 처리가 원활히 이루어 질 수 있도록 클래스 간의 자원을 동적으로 할당하는 메커니즘을 제안하였으며 제안한 메카니즘에 대한 시뮬레이션을 통한 성능 분석을 수행하였다.

An Efficient Admission Control and Resource Allocation Methods in DiffServ Networks

Byung-Il Cho[†], Sang-Jo Yoo^{**}

ABSTRACT

DiffServ network architecture does not define a call admission control procedure explicitly. In this paper, a new DiffServ QoS control mechanism is suggested which, after the call admission control, can execute packet process according to the class while, at the same time, executing on the flow based call admission control in the DiffServ network. Routers on the path from the source to the destination estimate the aggregated class traffic of the existing flows and the new incoming flow and then, perform a call admission control in accordance with the type of classes efficiently based on the required bandwidth per each class that can meet the user's QoS requirements. In order to facilitate the packet process according to the class after the flow based call admission control, a mechanism is suggested that can adjust the network resources to classes dynamically. The performance analysis on this mechanism is carried out through a simulation.

Key words: Internet QoS(인터넷QoS), DiffServ(차별화서비스), call admission control(호수락제어), resource allocation(자원할당)

1. 서 론

최근의 인터넷은 기존의 최선형 서비스구조 (best-effort)에서 벗어나 별도의 차별화 된 트래픽 처리와

서비스 품질을 요구하는 응용들을 지원할 수 있는 네트워크 구조 및 트래픽 엔지니어링 기술을 요구하고 있다. 따라서 인터넷 망 자원을 효율적으로 관리 하며 사용자 응용의 특성에 맞는 서비스 품질 (QoS:

※ 교신저자(Corresponding Author): 유상조, 주소: 인천광역시 남구 용현동 253번지(402-751) 전화: (032)872-8353, FAX: (032)865-0480, E-mail: sjyoo@inha.ac.kr

접수일: 2003년 5월 30일, 완료일: 2003년 10월 31일

[†] 정회원, 인하대학교 정보통신대학원 석사 과정

(E-mail: bigbang@igtel.co.kr)

^{**} 인하대학교 정보통신대학원 조교수

※ 본 논문은 인하대학교 교수연구진흥사업(30227-01) 지원으로 수행하였음.

Quality of Service)을 제공하기 위해서 사용자의 트래픽 변화에 따라 통신망을 효율적으로 관리하는 방안이 필요하다. 현재 IETF에서 활발히 논의되고 있는 QoS 제공 방법 중 하나가 클래스 별 차등 서비스(DiffServ: Differentiated Service) 구조이다. DiffServ 망에서의 QoS제공 패킷처리 메커니즘으로는 트래픽 셰이핑 (shaping), 미터링 (metering), 마킹 (marking) 및 스케줄링 (scheduling) 등이 있다. IETF의 또 다른 QoS 제공방법은 IntServ(Integrated Service)구조로, 플로우 별 호 수락 제어(call admission control)를 통해 플로우 단위 자원을 관리한다. 호 수락 제어의 목적은 망의 가용 자원을 수집 및 할당하여 가입자가 요구하는 QoS를 보장할 수 있는 신뢰성 있는 경로를 제공할 수 있도록 하는 것이다[1].

현재 인터넷 통신망에서 QoS 제공을 위한 호 수락 제어는 IntServ 구조에서 RSVP (Reservation Protocol) 프로토콜을 이용하여 가능하다[2-4]. IntServ에서는 RSVP 프로토콜을 이용하여 시그널링을 통해 패킷 손실률이나 지연 시간 등의 파라미터를 기준으로 호 수락 제어를 수행 하는데 이는 각 노드에서의 호 수락 제어 관리가 플로우 단위로만 이루어지기 때문에 절대적인 QoS를 보장할 수 있는 장점이 있으나 실제 통신망에 적용하기 위해서는 망 내 각 노드에서 호 수락된 각 연결들에 대한 파라미터 및 상태를 관리 하여야 하기 때문에 확장성 (scalability) 문제를 갖는다[5]. 이러한 IntServ의 문제점을 해결하기 위해 제안된 DiffServ는 IntServ의 플로우 별 QoS 제어와는 달리 망의 에지 라우터에서 해당 플로우에 필터링을 수행하여 플로우를 인식하고 SLA에 협약된 사항에 따라 DSCP (DiffServ Code Point) 값을 할당하고 사전에 협약된 트래픽 특성을 초과하여 입력되는가를 검사 (policing) 한다[6]. 즉 기존의

DiffServ 네트워크에서는 협약된 트래픽 특성을 유지하는 한 사용자 플로우는 모두 수락 되므로 명시적으로 플로우 별 호 수락 제어를 수행하지 않는다[7,8].

본 논문에서는 가입자의 트래픽 특성에 따른 정확한 플로우 별 호 수락 제어를 위해 DiffServ의 에지 라우터에서 플로우 단위의 트래픽을 클래스 별로 분류하기 전에 해당 플로우의 경로상의 라우터들이 적합한 서비스를 제공할 수 있는지 여부를 판단하는 과정을 통하여 호 수락 제어를 수행하는 메커니즘을 제안한다. 이 때 경로상의 라우터들은 클래스 별로 종합되어 있는 트래픽과 호를 요구하는 신규 플로우의 트래픽을 종합하여 도출한 소요대역을 이용하여 호 수락 여부를 결정한다. 본 논문에서는 사용자의 다양한 QoS 요구 조건에 따른 소요 대역 계산 방법 및 파라미터 도출 방법을 사용하여 클래스의 우선 순위에 따라 효율적으로 호 수락 제어를 할 수 있는 방안을 제시하였으며 호 수락 제어 후에도 클래스 별로 정적으로 디멘전닝 되어있는 링크의 이용 효율을 높이고, 플로우 기반의 호 수락 제어 후에 클래스 단위의 서비스 제공 시 발생할 수 있는 클래스 간의 부적절한 대역 할당 문제를 해결하기 위하여 클래스 간의 대역을 동적으로 할당하는 메커니즘을 제안하였다. 또한 사용자가 망에 접속하는 방법을 다양화 하여 최적화 된 트래픽 기술자를 이용한 호 수락 제어를 수행할 수 있도록 하였다. 표 1은 기존의 DiffServ와 본 논문에서 제시하는 DiffServ에서 호 수락 제어를 수행하는 메카니즘을 비교하여 정리하였다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 QoS 제공을 위한 호 수락 제어 메카니즘의 구조에 대해 살펴본다. 제 3 장에서는 DiffServ에서의 동적 대역 할당을 통한 호 수락 제어 프레임 워크를 제시하고 제 4 장에는 제안된 호 수락 제어 메카니즘의 수행

표 1. 기존의 DiffServ와 제안된 DiffServ에서의 호 수락 제어 메카니즘의 특징 비교

	DiffServ	DiffServ+호 수락 제어
호 수락 제어 여부	명시적으로 수행하지 않음	플로우 별 호 수락 제어를 수행하는 메카니즘을 적용
패킷 처리 방법	클래스 기반의 패킷 처리	클래스 기반의 패킷 처리 (호 수락 제어 시만 플로우 별 제어)
클래스 별 대역 할당	정적 대역 할당 (static)	동적 대역할당 (dynamic)+최소 대역보장 (static)
	망 사업자의 디멘전닝 설정에 의존	클래스 우선 순위에따라 대역 계산 및 파라미터 도출 방법이 세분화 됨
사용자의 접근 방법	SLA(Service Level Agreement)에 의존	가입자가 트래픽 전송 시 트래픽 기술자를 선언할 수 있도록 함 (선언하지 않을 경우 정책 서버의 SLA이용)

및 세부 실행 방안에 대해 설명한다. 제 5 장에서는 제안된 방법에 대해 성능 평가를 수행하였으며, 마지막으로 제 6 장에서 본 논문의 결론을 맺는다.

2. QoS 제공을 위한 호 수락 제어 메커니즘

호 수락 제어는 가입자가 SLA를 통해 미리 망과 협약된 서비스 품질을 제공 받기 위해 망에게 트래픽 전송 권한을 허가 받는 일련의 절차를 말한다. 사용자가 데이터 전송을 요구할 경우 전송할 트래픽의 특성을 기술하여 (기술하지 않을 수도 있음) 망에 호 수락을 요청하면 망에서는 사용자 플로우를 식별하는 과정을 거쳐 ((발신자 IP 주소, 수신자 IP 주소, 응용 프로토콜 종류)) 특정 클래스 사용 권한을 인식하고 종단점까지의 망의 노드-노드들이 사용자가 전송할 트래픽 특성에 대한 클래스 품질을 제공할 수 있는가를 검사한다. 만약 망의 가용 자원이 새로운 호 요구를 받아 들일 수 있으면 새로운 호는 수락되며, 그렇지 않을 경우 호는 거절 된다.

DiffServ에서의 호 수락 제어에 대한 연구는 크게 두 가지로 나뉘게 된다. 첫 번째는, 중앙 집중형 호 수락 제어로서 대역폭 브로커 (bandwidth broker)를 사용하여 가입자로부터 요구된 서비스에 대하여 수락 여부를 결정하고 자원을 할당한다. 여기서 새로운 호에 대하여 수락 제어를 하기 위해 중요한 것은 대역폭 브로커의 망 자원 관리 방법으로서 [9,10]에서 제안한 정적 자원 관리와 [11]에서 제안한 동적 자원 관리 방법이 있다. 정적 자원 관리의 사전에 정해진 라우팅 경로를 통하여 고정된 주기로 자원을 할당하고 새로운 호의 요청이 있을 때 호 수락 여부를 결정한다. 이는 망자원의 할당 변경이 일정기간 단위로 이루어짐에 따라 동적으로 변화하는 트래픽이나 라우팅 등의 망 토폴로지 변화에 즉각적인 대응이 어렵기 때문에 혼잡이 발생할 가능성이 있으며 망 자원의 낭비가 있을 수 있다. 동적 자원 관리 방법은 망의 혼잡 발생 시 링크의 상태를 대역폭 브로커에 보고하여 실시간으로 변화하는 트래픽이나 라우팅 등의 망 토폴로지 변화에 즉각적인 대응이 이루어 질 수 있지만 이 방법은 망 내 모든 노드에서 링크의 대역폭 사용량을 대역폭 브로커에 보고하여야 하며, 대역폭 브로커 또한 망의 대역폭 사용량에 대한 정보를 유지하여야 하고 각 노드로부터의 메시지 전송량이 많

질 경우 많은 부담이 있게 된다. DiffServ에서의 호 수락 제어를 위한 또 다른 방법은 측정 기반 호 수락 제어로서 이는 호가 요구하는 트래픽의 크기와 같은 가상의 탐색 패킷을 전송하여 경로 상의 각 노드가 새로운 호를 수용할 용량이 있는지를 측정하여 호 수락 제어를 수행하는 방법이다[11,12]. 이는 새로운 플로우에서 사용할 대역폭을 미리 망으로 전송하여 망의 상태를 탐색하는 것으로서 이를 위해서는 부가적으로 탐색 패킷을 위한 대역 관리 및 버퍼 관리 방법을 지원 해야 하고, 부가적 패킷의 생성으로 인하여 망에 추가적인 부담을 주게 된다. 결국 이러한 호 수락 제어 방법들은 네트워크의 자원을 효율적으로 이용하지 못하며 사용자에게 적절한 QoS를 제공하기에 어려움이 있으므로 DiffServ네트워크에서의 플로우 별 호 수락 제어를 위한 방안과 호 수락 제어를 능동적으로 수행할 수 있는 효율적인 네트워크 자원 할당 방안이 필요하다.

3. DiffServ 네트워크에서의 플로우 별 호 수락 제어 구조

본 절에서는 DiffServ에서 플로우 기반의 호 수락 제어 프레임 워크와 호 수락 제어를 위한 소요 대역 결정 방법을 제시하도록 한다.

3.1 DiffServ에서의 플로우 별 호 수락 제어 프레임 워크

본 논문에서는 DiffServ 망에서의 서비스 클래스를 Gold / Silver / Bronze의 세 가지로 구분 하여 클래스의 정책에 따라 호 수락 제어의 수행 여부를 결정하도록 하였다. 호 수락 제어는 Gold와 Silver 클래스에서만 수행 하도록 하며 Bronze 클래스는 최선형 서비스를 제공하는 클래스로 지정하여 호 수락 제어를 수행하지 않는다. 또한 본 논문에서는 사용자가 호 수락을 요구하는 시그널에 자신의 트래픽 형태에 대한 트래픽 기술자를 선언 하는 경우와 트래픽의 기술이 없이 단순히 호 수락 여부만을 질의하는 경우로 구분하였다. 사용자가 트래픽 기술자를 선언하는 경우에는 선언된 트래픽 기술자에 적합한 대역을 계산하여 호 수락 여부를 판단하고 트래픽 기술자를 선언하지 않는 경우에는 정책 서버로부터 SLA 정보를 받아 SLA에서 약정한 트래픽 정보를 호 수락 제

어에 이용하도록 한다. 호 수락 제어를 위해서는 트래픽 기술자 정보와 각 코어 노드에서의 호 수락 정보를 운반하는 프로토콜이 필요하다. 본 논문에서는 이러한 프로토콜을 DACS (DiffServ Admission Control Signaling) 이라고 명하도록 한다. DACS는 Int-Serv의 RSVP의 PATH 메시지와 RESV 메시지와 비슷한 패턴으로 동작하지만 호 수락 제어 시에만 동작하고 동작시에도 플로우의 트래픽 기술자 정보와 경로상의 코어 라우터들의 호 수락 정보만을 운반하기 때문에 RSVP에서처럼 경로 상의 모든 라우터에서 플로우에 대한 상태 관리 및 자원할당의 과정이 없어서 RSVP처럼 시스템에 부하를 주지 않는다. 그림 1은 호 수락을 요구하는 플로우에 대한 DACS가 이동하면서 호 수락 제어를 수행하는 과정이다.

사용자는 데이터 전송을 위해 네트워크에 호 수락을 요구하는 시그널을 입구 라우터 (ingress router)에 보내게 된다(그림 1 - ①). 호 수락 요구 시그널을 받은 입구 라우터는 호 수락 요구 시그널에 포함되어 있는 트래픽 기술자 (또는 트래픽 기술자가 포함되어 있지 않을 경우 정책 서버로부터 받은 SLA정보(그림 1 - ①'))를 포함한 RD 메시지 (resource dis-

covery message)를 라우팅 경로상의 코어 라우터 (core router)로 보내게 된다 (그림 1 - ②). RD 메시지는 RSVP의 PATH와 유사하고 Tspec에 해당하는 트래픽 정보인 리키 버킷 기반의 트래픽 기술자 정보를 가지게 된다. RD 메시지를 받은 코어 라우터는 현재 망의 가용 대역 자원을 계산하여 새로운 호가 요구하는 대역을 수용할 수 있는지를 비교하여 호 수락 여부를 결정한 후 RD 메시지에 이를 표시하여 다음 라우터로 전송하게 된다. 이 과정에서 코어 라우터에서의 대역 계산 시 현재 서비스 중인 기존호의 트래픽과 호를 요구하는 신규 호의 트래픽을 중합하여 소요 대역을 구하고 이를 호 수락 제어에 사용한다 (그림 1 - ③). 출구 라우터 (egress router)까지 RD 메시지의 전송이 끝나게 되면 송신 방향으로 RSVP의 RESV 메시지에 해당하는 응답 메시지 (reply message)를 보내어 전체 경로의 호 수락 정보를 입구 라우터에 전송하고 (그림 1 - ④), 입구 라우터는 호 수락 여부를 결정하게 된다 (그림 1 - ⑤). 이와 같은 신규 플로우에 대한 호 수락 제어를 수행한 후에는 기존의 DiffServ 메커니즘과 동일한 클래스 별 대역 할당 및 스케줄링이 이루어지게 되며 클래스

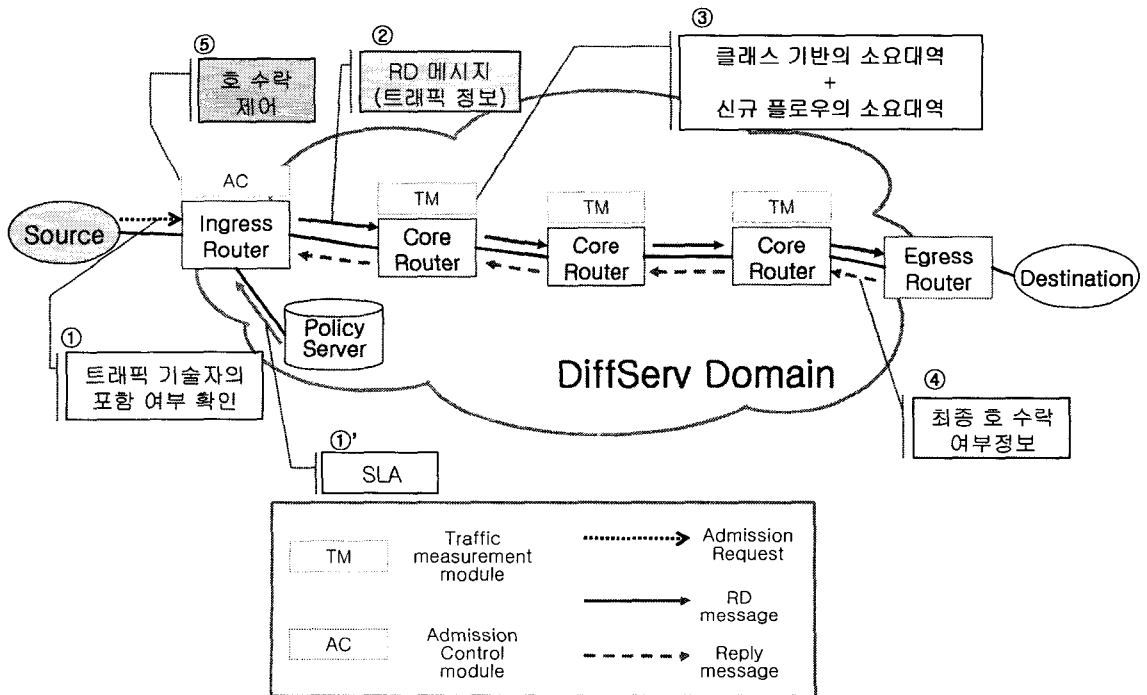


그림 1. DiffServ에서의 호 수락 제어 구조

기반의 PHB (Per Hop Behavior)가 수행된다. 호 수락 제어 시 해당 플로우가 속한 클래스의 소요 대역 계산은 Gaussian 모델을 이용하며 계산에 필요한 파라미터인 트래픽의 (평균, 분산)은 트래픽 기술자를 이용한 해석적 방법이나 일정 주기의 트래픽 관측을 통하여 도출 하도록 한다. 그림 2는 DiffServ에서의 호 수락 제어를 위한 DACS 프로토콜의 흐름도 이다.

3.2 클래스 별 트래픽 소스 모델링 및 실효 대역 추정 방법

각 노트에서 호 수락 여부를 결정하기 위해 필요한 서비스 클래스별 링크 용량을 구하기 위해서 필요한 트래픽 소스 모델링에는 다음의 방법들이 사용될 수 있다.

첫 번째는, 유한 버퍼 길이 시스템에서 Markov 모델을 이용한 실효대역(\hat{C}) 계산 방법으로 버퍼길이 가유한한 x 일 경우 다음과 같은 식 (1)을 이용한다[13].

$$\varepsilon = \delta \times \exp\left(-\frac{x(\hat{C} - \rho \times PDR)}{b(1-\rho)(PDR - \hat{C})\hat{C}}\right), \tag{1}$$

$$\delta = \frac{(\hat{C} - \rho \times PDR) + \varepsilon \rho (PDR - \hat{C})}{(1-\rho)\hat{C}}$$

여기서 $\varepsilon = PLR$, $b = 1/\beta$, $\rho = \alpha/(\alpha + \beta)$ 이며 α, β 는

각각 ON/OFF 모델에서 ON 상태 및 OFF 상태 이탈률을 의미하며, 리키버킷 기반 트래픽 기술자로부터 식 (2)와 같이 도출된다.

$$\alpha = \frac{PDR \times SDR}{MBS(PDR - SDR)} = \frac{1}{E[OFF]},$$

$$\beta = \frac{PDR}{MBS} = \frac{1}{E[ON]},$$

두 번째로, 무한 버퍼 시스템에서 Markov 모델을 이용한 실효대역 계산 방법으로 이러한 방법은 단일 소스 응용의 일정 지연이 있는 클래스에 적합하며 각 개별 소스 i 의 소요 대역 (C_i)는 식 (3)과 같이 구해진다[13]. 무한 버퍼 시스템에서 x 는 최대지연 요구 사항에 대응되는 버퍼 길이로 x 를 초과하여 입력된 패킷은 손실로 고려된다.

$$C_i = \frac{\omega b(1-\rho)PDR - x + \sqrt{[ab(1-\rho)PDR - x]^2 + 4x\omega b\rho(1-\rho)PDR}}{2\omega b(1-\rho)},$$

$$\omega = \ln \frac{1}{\varepsilon} \tag{3}$$

세 번째로, 버퍼가 없는 시스템에서 Gaussian 모델을 이용한 소요 대역 산출 방법으로 ON/OFF 모델 소스 다수 개가 다중화 되면 Gaussian 분포를 따르게 된다고 가정하고 이를 근거로 패킷 지연이 없는

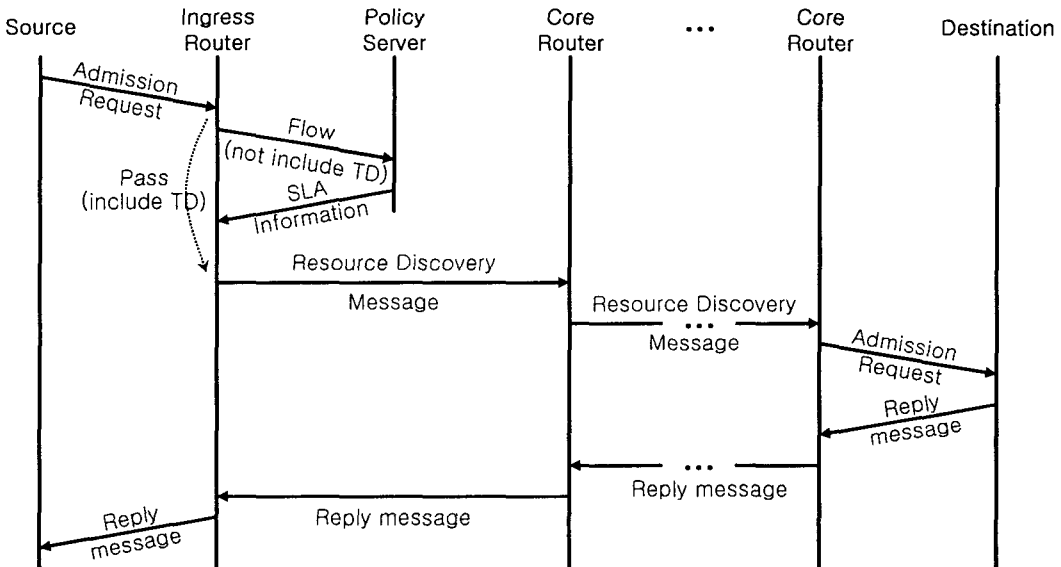


그림 2. DiffServ에서의 호 수락 제어를 위한 DACS 프로토콜의 흐름

조건에서 패킷 손실율만을 고려한 클래스 k 의 소요 대역 (C_R^k)을 도출하는 방법이다. 이 방법은 저 지연, 저 손실의 고품질 QoS를 요구하는 우선 순위가 높은 클래스 (예: Gold 서비스 클래스)에 효과적으로 적용될 수 있으며 식 (4)와 같이 구할 수 있다[14]. 여기서 m_k 와 σ_k 는 각각 클래스 k 의 중합된 트래픽의 평균과 표준 편차이다. 또한 ε 는 요구된 패킷 손실률이다.

$$C_R^k = m_k + \gamma \times \sigma_k$$

$$\gamma = \sqrt{-2 \ln \varepsilon - \ln(2\pi)} \quad (4)$$

네 번째로, 무한 버퍼 시스템에서 Gaussian모델을 이용한 소요 대역 계산 방법으로 버퍼에 입력되는 트래픽의 특성을 고려한 대역 할당을 위해 식 (5)와 같이 자기상관함수 감소율 계수(η)를 사용한다. 이 방법은 어느 정도의 패킷 지연과 패킷 손실이 허락되는 클래스(예: Silver 및 Bronze 서비스)에 적용 가능하다[14]. 여기서 x 는 버퍼 길이이다.

$$C_R^k = m_k + \gamma \times \sigma_k$$

$$\gamma \approx \sqrt{d - 2 \ln \left(\sqrt{d} - \frac{\eta}{\sigma} x \right) - \frac{\eta}{\sigma} x}, \quad (5)$$

$$d = -2 \ln(\sqrt{2\pi} \times \varepsilon) + \frac{\eta^2}{\sigma^2} x^2$$

위 방법들 중 어떤 방법이 어느 클래스에 적합한가는 가입자 수, 트래픽의 특성, QoS 요구 사항 조건 등에 따라 달라진다[15].

3.3 호 수락 제어를 위한 소요 대역 결정 방법

코어 라우터들은 에지 라우터로부터 받은 신규 플로우의 트래픽 기술자 정보를 바탕으로 서비스에 필요한 소요 대역을 계산하여 호 수락 여부를 결정한다. 본 절에서는 코어 라우터에서 호 수락 여부를 판단할 때 필요한 소요 대역을 계산하는 방법에 대해 기술 한다.

본 논문의 코어 라우터에서의 호 수락 여부 결정에서는 현재 서비스 중인 클래스 기반의 트래픽들과 신규 플로우의 예측되는 트래픽을 중합하여 계산한 총 대역 (C_R^k)을 이용한다. 여기서 k 는 클래스 식별자이다. 이를 위해 본 논문에서는 앞절에 기술된 트래픽의 소요대역 계산방법 중 기존 트래픽과 신규 트래픽의 중합이 용이하고 계산 과정과 파라미터 도출이

비교적 간단한 Gaussian 모델을 이용한 소요 대역 계산 방법을 사용한다. 이러한 Gaussian모델을 이용한 소요 대역 계산 방법은 트래픽의 {평균, 분산}이 필수적인 파라미터이다. 따라서 본 논문에서는 트래픽의 {평균, 분산} 도출 방법을 다음의 두 가지 방법으로 구분 하여 클래스 우선 순위 따라 선택할 수 있도록 고려하였다.

{평균, 분산}을 도출하는 방법의 첫 번째는 해석적 방법[14]으로서 다중화된 클래스의 각 플로우의 리키버킷 기반 트래픽 기술자 (또는 SLA)를 이용하여 중합된 평균과 분산을 구한다. 해석적 방법에서 먼저 식 (6)을 이용하여 각 플로우의 평균과 분산을 구하고 식 (7)을 이용하여 중합된 클래스 전체의 평균과 분산을 구한다.

$$m_i = SDR_i, \quad \sigma_i^2 = SDR_i(PDR_i - SDR_i) \quad (6)$$

$$m_k = \sum_{i=1}^N m_i, \quad \sigma_k^2 = \sum_{i=1}^N \sigma_i^2 \quad (7)$$

여기서 사용하는 PDR_i (Peak Data Rate), SDR_i (Sustainable Data Rate)은 트래픽 기술자(또는 SLA 정보)파라미터이다. 결국, 해석적 {평균, 분산} 도출 방법은 트래픽 기술자로부터 도출 되므로 트래픽의 QoS를 보장하는 상위 경계값(upper bound)으로 도출된다. 따라서 고품질의 QoS를 요구하는 우선 순위가 높은 클래스의 소요대역 계산에 이용한다.

{평균, 분산}을 도출하는 두 번째 방법은 관측에 기반한 방법으로서 관측은 N_s 개의 관측 구간에 대해 수행된다. 단 무한 버퍼에서의 $P(Q > x) = \varepsilon$ 개념의 지연과 손실 QoS를 고려한 소요대역을 계산하기 위해서는 autocovariance의 추정도 필요하다. 관측은 매 1분마다 실시하였으며 실제 시스템에서도 식 (8)을 이용하여 간단한 형태의 {평균, 분산} 관측 방법이 사용가능하다.

$$m_k = \frac{1}{N_s} \sum_{n=1}^{N_s} R_n$$

$$\sigma_k^2 = \frac{1}{N_s} \sum_{n=1}^{N_s} (R_n - m_k)^2$$

$$Cov(y) = \frac{1}{N_s - y} \sum_{n=1}^{N_s - k} (R_n - m_k)(R_{n+k} - m_k), \quad (8)$$

$$y = 0, 1, 2, \dots, N_s - 1$$

현재 클래스의 수락된 트래픽의 특성을 관측을 통해 전체 평균(m_k)과 분산(σ_k^2)을 구한다. 이러한 관측에 의한 {평균, 분산} 도출 방법은 평균적 관점의 트래픽 {평균, 분산}이 도출 되므로 비교적 유연한 QoS를 제공하는 우선순위가 낮은 클래스에 적용 가능하다. 결국 신규 호와 기존 호의 {평균, 분산}을 종합한 $\{m_k, \sigma_k^2\}$ 의 값은 다음 식 (9)와 같이 구한다.

$$\begin{cases} m_k = m_k^{old} + m_N \\ \sigma_k^2 = \sigma_k^{2old} + \sigma_N^2 \end{cases} \quad (9)$$

여기서 해석적 방법을 이용할 경우 기존 호들의 $\{m_k^{old}, \sigma_k^{2old}\}$ 는 식 (6)과 (7)을 이용하여 구하고 관측적 방법을 이용할 경우 기존 트래픽의 관측을 통해 구한다. 단, 두 방법 모두 신규 호의 평균과 분산 $\{m_N, \sigma_N^2\}$ 은 사전 관측이 불가능 하므로 식 (6)의 해석적 방법

을 이용한다.

이와 같이 각 서비스 클래스의 우선 순위에 적합한 소요 대역 계산 방법 및 트래픽 파라미터 도출 방법을 선택하여 해당 클래스의 특성에 맞는 최적의 링크 용량을 도출 한다. 코어 라우터에서의 호 수락 제어 결정에 필요한 다중화 된 클래스 트래픽 파라미터 $\{m_k, \sigma_k^2\}$ 을 얻은 후에 클래스 k 의 소요대역을 식 (4) 또는 식 (5)를 이용하여 구한다.

그림 3은 코어 노드에서 신규 호의 수락 여부를 결정하기까지의 과정이다. 기존의 서비스 중인 트래픽과 신규 트래픽의 {평균, 분산}을 클래스의 우선순위에 적합한 방법으로 구한 후 소요 대역 계산 방법 역시 클래스의 우선순위에 적합한 방법을 선택하여 구한 값을 이용하여 현 노드에서의 호 수락 여부를 결정한다. 본 논문에서 제안하는 서비스 클래스의 우선 순위에 따른 소요 대역 계산 방법과 소요 대역

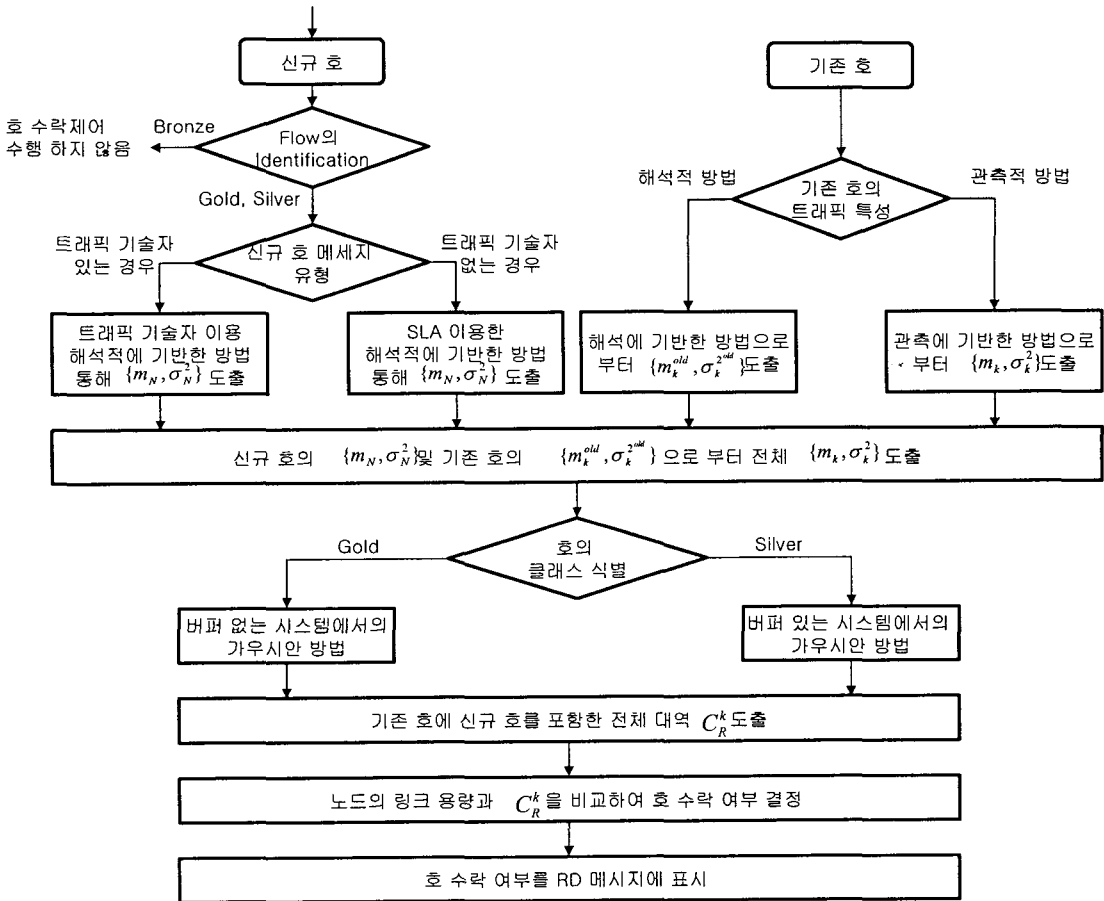


그림 3. 코어 노드에서의 대역 계산 과정 및 호 수락 제어 결정 과정

제산에 필요한 파라미터의 선택은 표 2와 같다.

4. QoS제공을 위한 효율적 자원 관리 메카니즘

본 절에서는 클래스 별 소요 대역 결정 방법을 이용하여 계산된 실제DiffServ 네트워크의 대역을 이용하여 호 수락 제어 여부를 결정하는 방법과 각 노드의 링크 대역 자원을 효율적으로 사용할 수 있는 자원 관리 방법에 대해 기술하도록 한다.

클래스 k 의 전체 링크 소요 대역을 C_R^k 라고 할 때 가능한 호 수락 제어 방법에는 다음의 두 가지가 있다. 첫 번째는 망의 디멘전닝 시에 할당된 클래스 별 대역(C_{dim}^k)을 기준으로 식 (10)과 같은 범위에서 호를 수락 제어 하는 방법이다. 이러한 방법은 정적인 대역 할당으로 인하여 전체 시스템 구성이 간단하며 엄격한 QoS 보장이 가능하다는 장점이 있지만 망의 디멘전닝 시와는 다르게 특정 클래스로 트래픽이 집중 될 경우 해당 클래스에 대해서는 호 거부가 증가하고 트래픽이 적게 발생하는 클래스에 대해서는 대역폭의 낭비가 심하게 되어 유연성 있는 대처가 어렵다는 단점이 있다.

$$\begin{cases} C_R^k \leq C_{dim}^k, & \text{call accept,} \\ C_R^k > C_{dim}^k, & \text{call reject.} \end{cases} \quad (10)$$

따라서 본 논문에서는 대역폭을 보다 효율적으로 사용할 수 있으며 보다 많은 가입자에게 원하는 QoS를 제공할 수 있는 방법을 제안한다. 이는 식 (11)과 같은 기준에서 호를 수락 제어하는 방법으로, 이는 전체 링크 용량 C 를 유연하게 사용하기 위해서 전체 링크의 대역폭과 비교하여 각 클래스들의 요구 대역의 합이 작으면 일단 호를 수락 한 후에 여유가 있는 클래스의 대역을 대역이 부족한 클래스에게 동적으로

할당하는 방법이다. 여기서 C_C^k 은 클래스 k 의 요구 대역에서 클래스 별 동적 대역 할당을 통해 변경된 대역(CR: compensated required capacity)을 의미한다.

$$\begin{cases} \sum_{k=1}^n C_{CR}^k \leq C, & \text{call accept, } (n: \text{The number of classes}) \\ \sum_{k=1}^n C_{CR}^k > C, & \text{call reject.} \end{cases} \quad (11)$$

C_{CR}^k 의 사용은 기본적으로 디멘전닝된 것보다 작을 경우 C_R^k 을 갖는 클래스의 자원을 디멘전닝 보다 큰 자원을 요구하는 클래스로 대역 사용권을 넘겨주기 위해 사용된다. 그러나 대역폭의 유연성을 강조하다 보면 우선순위가 높은 클래스의 가입자보다 우선순위가 낮은 클래스의 가입자가 미리 대역을 점유하여 블로킹 현상이 발생 할 수 있으므로 본 논문에서는 이를 방지하기 위해서 각 클래스 별 최소 대역폭인 C_{min}^k 을 제안한다. C_{min}^k 은 클래스 k 에 최소한 무조건 할당되는 대역폭으로 다음의 식 (12)의 범위 내에서 결정된다. 최소 대역의 용량은 클래스 우선 순위가 높을 수록 많은 할당량을 배분하도록 한다. 최소 대역폭의 세부 값은 사업자의 정책에 의한 클래스 별 우선 순위에 따라 다양하게 정할 수 있다.

$$0 \leq C_{min}^k \leq C_{dim}^k \quad (12)$$

결국 클래스 별 요구 링크 용량은 다음의 식 (13)와 같이 구해진다.

$$C_{CR}^k = \max \{C_{min}^k, C_R^k\} \quad (13)$$

이러한 동적대역할당은 기본적으로 특정 클래스의 소요대역이 이미 디멘전닝되어 있는 대역을 초과하였을 때 작동한다. 또한 확장된 클래스의 대역폭은

표 2. 서비스 클래스 별 링크 용량 산출을 위한 계산 방법 및 파라미터

서비스 클래스	소요 대역 계산 방법	파라미터 도출방법	트래픽 기술자 (또는 SLA)	호 수락 제어
Gold	버퍼 없는 시스템의 Gaussian모델	해석적 방법	PDR, SDR, MBS	수행
Silver	무한 버퍼 시스템의 Gaussian모델	해석적 방법	PDR, SDR, MBS	수행
Bronze	무한 버퍼 시스템의 Gaussian모델	관측적 방법	가입 시 트래픽 기술자를 선언하지 않음	수행 안 함

MBS (Maximum Burst Size)

디멘전닝 보다 크게 사용하고 있는 클래스(상대방의 최소대역폭은 침범하지 못함)의 소요대역이 작아져서 자신의 디멘전닝 대역보다 작아지면 대역폭이 완전히 반납되게 된다. 이 때 클래스별 소요대역(C_{CR}^k)을 근거로 대역을 조정하고 스케줄링 정책에 따라서 입력 파라미터를 변경하게 된다.

제안된 방법은 클래스 별 대역폭을 동적으로 변화 시키기 때문에 PHB의 스케줄링 파라미터 역시 그에 맞추어 동적으로 변화 시켜야 한다(그림 4-(a)). (그림 4-(b))는 클래스 별 대역폭을 동적으로 할당하는 단계를 예를 들어 나타낸 것이다. (그림 4-(b)(i))은 망 사업자가 디멘전닝에 의해 전체 링크 용량을 클래스 별로 분배하였음을 보여준다. (그림 4-(b)(ii))는 클래스 별 최소 대역폭을 나타낸다. (그림 4-(b)(iii))은 각 클래스 별로 들어오는 트래픽을 나타낸 것으로서 Silver 클래스에는 디멘전닝 시 보다 많은 용량의 트래픽이 들어오고 있으며 Gold 클래스에는 대역폭의 여유가 있음을 나타낸다. 이러한 경우 본 논문에서 제안하는 동적 대역폭 할당 메카니즘을 이용하여 (그림 4-(b)(iv))와 같이 Gold 클래스의 여분의 대역폭을 Silver 클래스로 할당하여 전체 링크 용량의 관점에서 보았을 경우 대역폭의 이용률과 효율성을 높이게 된다. 단, 클래스 별 최소 대역폭은 나중에 대역폭의 여유가 있던 우선 순위가 높은 클래스(Gold)에 트래픽이 몰릴 경우 우선 순위가 낮은 클래스에 비해 호 수락율이 감소되는 것을 방지 하기 위해 사용된

다. 이러한 동적 대역할당은 특정 클래스의 소요대역이 이미 디멘전닝되어 있는 대역을 초과하였을 때 사용되며, 대역폭 조정을 위하여 클래스별 소요대역(기호)을 입력 파라미터로 이용하고, 대역 조정 이후에는 스케줄링 정책 및 파라미터가 변경된다. 또한 확대된 대역폭은 디멘전닝 보다 크게 사용하고 있는 클래스(상대방의 최소 대역폭은 침범못함)의 소요대역이 작아져서 자신의 디멘전닝 대역보다 작아지면 대역폭이 완전히 반납되어 축소된다.

그림 5는 본 논문에서 제안하는 클래스 별 동적 대역 할당을 통한 플로우 별 호 수락 제어 메카니즘의 흐름도 이다. 그림 5에 따르면 신규 호와 기존 호에 각각 적합한 소요 대역 계산 방법을 이용하여 전체 소요 대역을 구한다. 그 다음 플로우 별 호 수락 및 링크의 효율성을 고려하여 고안된 클래스 별 동적 대역 할당 메카니즘을 이용하여 대역의 동적 할당을 수행 한다. 이 때 우선순위 클래스의 QoS 보장을 위해 최소 대역폭을 설정 할 수 있다. 이러한 과정을 거쳐 최종적으로 호 수락 제어 여부를 판단하여 RD 메시지에 표시 하게 된다. RD 메시지는 현재 노드에서 호가 수락되었으면 다음 노드로 향하게 되고 호 거부 메시지를 받았을 때는 송신지로 되돌아 가게 된다. RD 메시지가 출구 라우터까지 도착하면 출구 라우터는 응답 메시지를 보내게 되는데 각 코어 라우터들은 이 응답 메시지를 통하여 최종적으로 호 수락 사실을 인지 하게 된다.

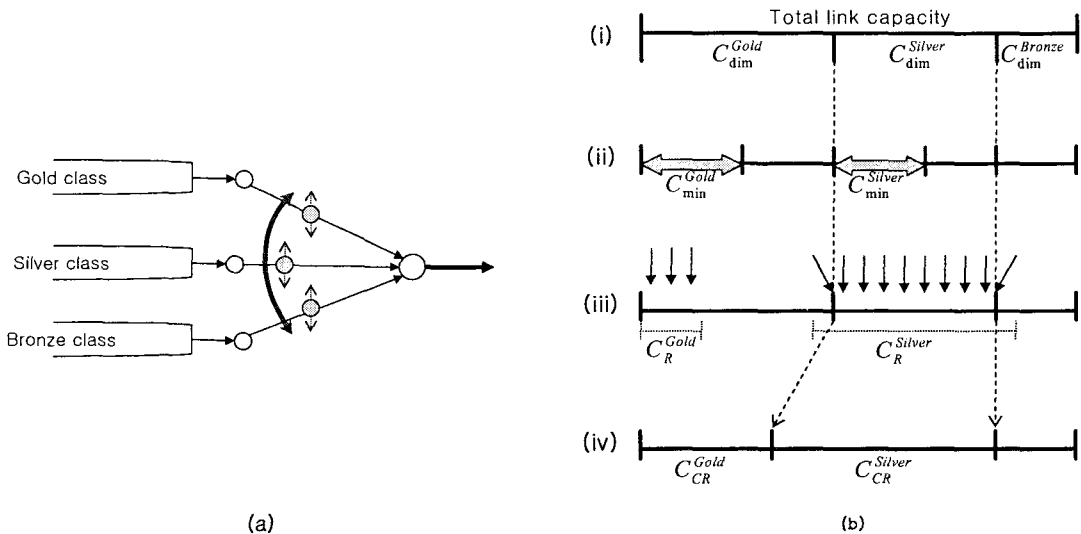


그림 4. 클래스 별 동적 대역 할당의 예

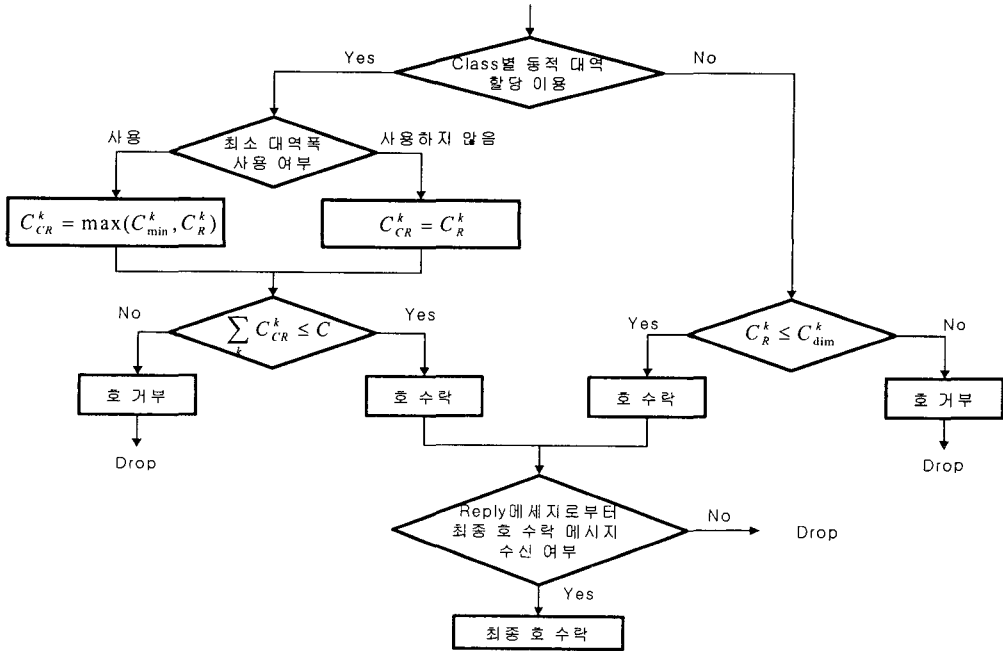


그림 5. 클래스 별 동적 대역 할당을 통한 플로우 별 호 수락 제어 메카니즘의 흐름도

이와 같은 DiffServ에서의 플로우 별 호 수락 제어를 위해서 본 논문에서는 클래스 우선 순위에 따라 소요 대역 계산 방법을 구분하였고, 소요 대역 계산에 필요한 파라미터 도출 방법 역시 세분화 하였으며, 링크의 이용 효율 증대 및 원활한 트래픽 전송을 위해 제안한 동적 대역 할당 후에도 우선 순위 클래스의 QoS 보장을 위해서 최소 대역폭을 정의 하는 메카니즘을 도입하였다. 이러한 소요 대역 계산 방법, 파라미터 도출 방법, 최소 대역폭 등의 실행을 위한 세부 파라미터는 그림 6과 같은 클래스 우선 순위에 따라 망 사업자가 선택할 수 있다

High	해석적 방법	버퍼 없는 Gaussian	다변량 정규 시 대역폭
↑ Class Priority ↓	↑	↑	↑
	↓	↓	↓
Low	관측에 의한 방법	무한 버퍼 Gaussian	0
	파라미터 추정 방법	소요 대역 계산 방법	최소 대역폭

그림 6. DiffServ에서 호 수락 제어를 위해 제안한 방법들의 수행 기준

5. 모의 실험 및 성능 분석

본 장에서는 제안된 DiffServ 네트워크에서의 호 수락 제어 메카니즘 및 링크 용량 산출 방법을 기반으로 실험을 수행 하였다. 해석적 파라미터 도출 방법에 기반한 ‘링크 용량 산출 도구’를 개발하였으며 이를 이용하여 계산된 링크 별 서비스 클래스 소요 용량에 대하여 링크 용량 성능 평가 시뮬레이터를 구현하여 다양한 형태의 트래픽을 실제로 발생시켜 얻은 결과를 비교하여 본 논문에서 제안한 DiffServ 네트워크에서의 호 수락 제어 방법의 효율성과 타당성을 검증하였다. 시뮬레이션에 필요한 입력 파라미터는 표 3과 같으며 이를 이용하여 개별 플로우의 소스 트래픽을 발생시켰으며 패킷의 길이는 지수 합 수 분포를 갖도록 하였다. 본 실험은 단일 노드 링크에서 진행 되었으며 사용된 각 파라미터 역시 단일 노드의 링크를 기준으로 하였다.

본 논문에서는 Bronze 클래스는 최선형 서비스를 제공하는 서비스로 설정하여 호 수락 제어를 수행하지 않는다고 가정하였다. 따라서 Gold와 Silver 서비스 클래스에 대한 실험을 수행 하였으며 두 클래스에 대해 수행한 구체적인 실험 내역은 다음과 같다.

표 3. 실험을 위한 서비스 클래스 별 파라미터

서비스 클래스	SLA 트래픽 기술자	파라미터 값
Gold클래스	PDR(kbits/sec) SDR(kbits/sec) MBS(kbits) 단일 노드 목표 손실율: PLR 평균 패킷 크기: APS	PDR= 1000 kbps SDR= 500 kbps MBS= 200 kbits PLR= 0.001 APS= 3000bits
Silver클래스	PDR(kbits/sec) SDR(kbits/sec) MBS(kbits) 단일 노드 목표 손실율: PLR 평균 패킷 크기: APS	PDR= 1000 kbps SDR= 500 kbps MBS= 200 kbits PLR= 0.001 APS= 3000bits

- 클래스 별 소요 대역 계산을 위한 파라미터(클래스의 증합된 평균과 분산) 도출 방법인 해석적 방법과 관측에 기반한 방법에 따라 각각 새로운 호를 수락할 수 있는 대역을 계산하여, 트래픽을 발생 시킨 후 각각 호 수락 되는 고객 수를 비교하고, 각 방법에서 트래픽의 평균 지연 시간과 손실률을 실험을 통해 도출하여 각각 원하는 QoS의 보장 여부를 측정한다.
- 링크 이용의 효율성을 증가시켜 가능한 많은 고객에게 서비스를 제공하기 위한 클래스간 동적 대역 할당 메커니즘을 적용한 실험을 수행하고, 우선 순위 클래스의 QoS를 보장하기 위한 최소 대역폭을 적용하여 대역 이용 효율성을 분석한 후 적합한 최소 대역폭의 범위를 예를 들어 실험한다.

5.1 해석적 방법과 관측적 방법의 파라미터 도출 방법에 따른 호 수락 제어 비교

첫 번째 실험으로, 클래스 별 소요 대역 계산을 위한 파라미터 도출 방법인 해석적 방법과 관측에 의한 방법에 따라 호 수락 되는 고객 수를 비교 하였다. Gold 클래스 트래픽을 발생 시켜 호를 요구하는 고객에 대하여 해석적 방법으로 구한 {평균, 분산}을 이용하여 계산한 소요 대역 한도 내에서 수락되는 고객의 수와 실제 사용자의 트래픽 전송률이 계약(또는 약속)한 트래픽의 70%일 때 관측적 방법으로 구한 {평균, 분산}을 이용하여 계산한 소요 대역 한도 내에서 호를 요구하는 고객에 대하여 수락되는 고객의 수는 그림 7과 같다. 그림 7의 결과에서처럼 관측적 방법에 의해 도출한 파라미터를 이용한 계산 방

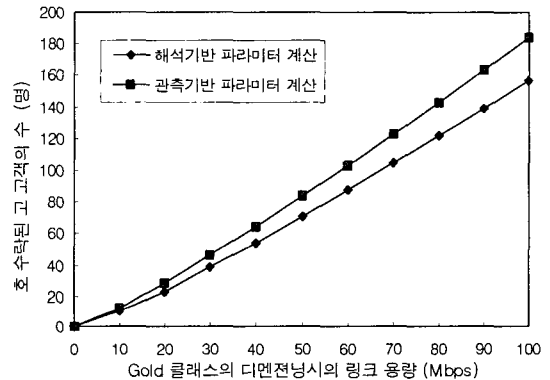


그림 7. 동일 조건에서의 대역 계산 방법파라미터 도출 방법에 차이에 따른 수락된 고객 수의 차이비교

법이 해석적 방법에 의해 도출한 파라미터를 이용한 계산 방법 보다 많은 수의 호를 수락할 수 있음을 알 수 있다. 해석적 방법은 고객이 보내는 호 요구 메시지에 포함된 트래픽 기술자나 사전에 계약된 SLA에 근거하여 파라미터를 계산한다. 이는 망 사업자 입장에서의 최악의 경우를 가정한 파라미터 계산 방법이다. 그러나 실제 네트워크에서 모든 고객이 항상 약정된 레이트로 접속 시간 동안 트래픽을 발생시키지 않으므로 고객의 실제 트래픽 전송률에 따라 관측 기반 트래픽 파라미터 계산은 해석적 기반의 대역 계산의 결과 보다는 작게 된다. 따라서 해석적 방법의 파라미터를 이용하여 계산한 소요 대역이 관측적 방법의 파라미터를 이용하여 계산한 소요 대역보다 크게 계산된다.

다음으로 해석적 파라미터 도출 방법과 관측적 파라미터 도출 방법에 의해 호 수락된 패킷에 대하여 QoS의 품질 평가를 위해 평균 지연 시간과 평균 손

실율을 측정하였다. 이의 결과는 각각 그림 8과 그림 9와 같다. 두 방법 모두 목표로 하는 패킷 손실율과 지연 시간의 기준을 충분히 준수하여 Gold 클래스 고객이 원하는 QoS를 엄격히 준수할 수 있음을 증명한다. 이상의 실험 결과 해석적 방법보다는 관측에 의한 방법이 제한된 용량에서 호 수락 고객을 늘릴 수 있었으며, QoS도 목표치를 준수할 수 있음을 알 수 있다. 그러나, 엄격한 QoS를 요구하는 클래스나 관측에 따른 오버헤드를 줄이고자하는 경우는 해석 파라미터에 근거한 대역 계산방법이 바람직함을 알 수 있다.

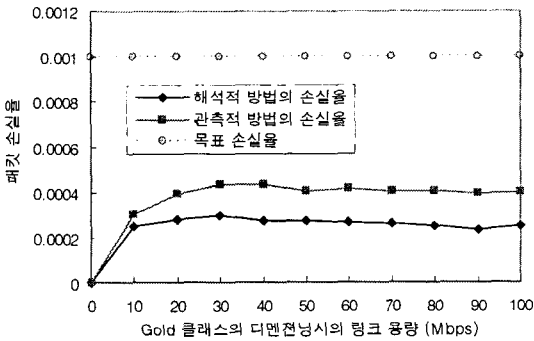


그림 8. 파라미터 도출 방법에 따른 패킷의 손실율

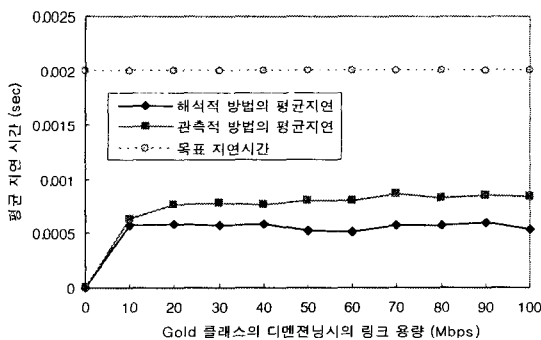


그림 9. 파라미터 도출 방법에 따른 패킷의 평균 지연시간

5.2 동적 대역 할당 방법의 성능 평가

다음 실험은 링크 이용의 효율성을 증가시켜 가능한 많은 고객에게 서비스를 제공하기 위해 본 논문에서 제안한 클래스간 동적 대역 할당 메카니즘과 최소 대역폭을 실제 실험에 적용하여 대역 이용 효율의 우수성을 검증할 수 있었다. 실험에 필요한 기본 클래스 별 파라미터는 표 2와 같다. 본 실험에 필요한

추가 파라미터는 다음과 같다. 망 디멘전닝 시 링크에 할당된 대역폭은 50Mbps이며 이 중 Gold 클래스에 30Mbps, Silver 클래스에 20 Mbps가 할당되어 있다고 가정한다. 이를 해석적 기반의 파라미터 도출 방법 및 표 1의 소요 대역 계산 방법에 근거하여 고객이 표 2의 설정으로 트래픽을 발생 시킨다고 가정 하였을 경우 Gold 클래스는 38명의 고객에게 서비스가 가능하고 Silver 클래스는 30명의 고객에게 서비스가 가능하다. 그러나 트래픽은 Gold 클래스 고객이 디멘전닝 시보다 적은 수가 호를 요청하고 Silver 클래스로는 디멘전닝 시 보다 많은 고객이 호를 요청하는 경우를 가정하여 실험을 수행 하였다. 동적 대역폭 할당 시의 결과 비교를 위해 먼저 그림 10에 동적 대역폭을 할당하지 않은 경우의 결과를 나타내었다.

그림 10을 보면 Silver 클래스에는 디멘전닝 시 30명의 고객을 수용하도록 설계되었지만 실제로는 50명이 넘는 고객이 호를 요청하고 있는 반면에, Gold 클래스는 디멘전닝 시 38명의 고객을 수용하도록 설계되었지만 실제로는 13명 이하의 고객이 호를 요청하여 충분한 QoS를 제공하면서 서비스를 제공하기도 여분의 대역폭이 많이 남아 있음을 알 수 있다. 이러한 경우 본 논문에서 제안하는 동적 대역 할당 방법을 사용하여 Gold 클래스의 여분의 대역폭을 Silver 클래스에 동적으로 할당한다. 단, 이때 Gold 클래스의 최소한의 QoS 보장을 위하여 최소 대역폭을 여분으로 남겨 두도록 한다. 그림 11은 최소 대역폭을 $C_{min}^k = C_{dim}^k / 2$ 로 설정 하여 Gold 클래스에 최소 대역폭을 15Mbps 남겨둔 후의 여분인 15Mbps를

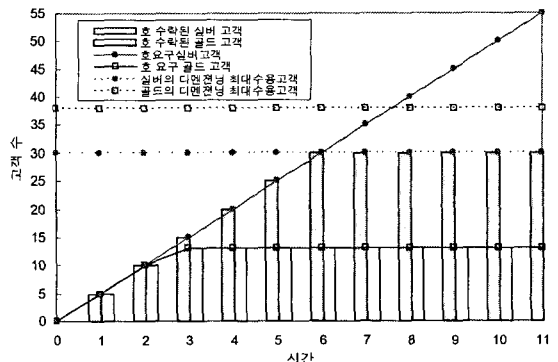


그림 10. 동적 대역폭을 할당하지 않은 경우의 클래스 별 호 수락된 가입자 수

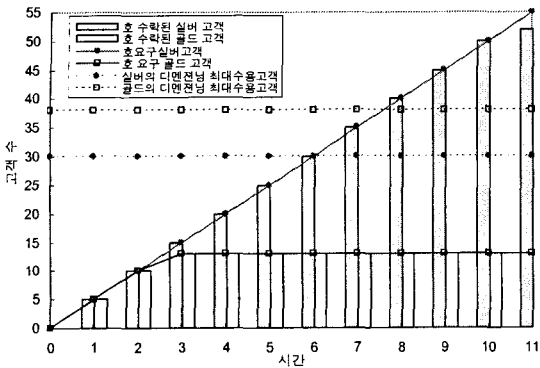


그림 11. 골드 클래스의 최소 대역폭을 $C_{min}^k = C_{dm}^k / 2$ 로 설정하여 동적 대역폭 할당

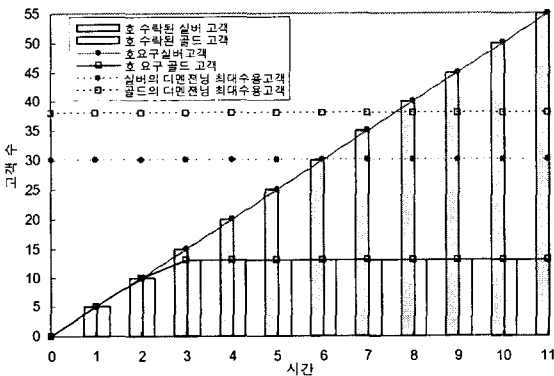
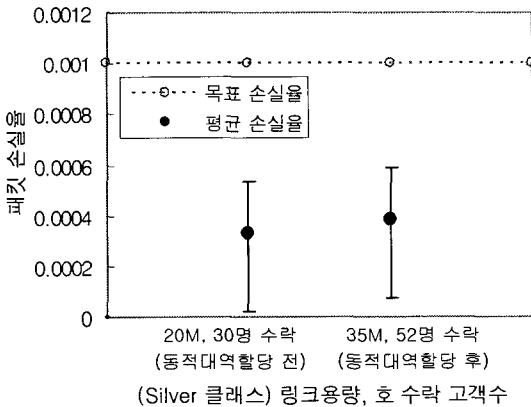


그림 12. 골드 클래스의 최소 대역폭을 $C_{min}^k = 0$ 로 설정하여 동적 대역폭 할당

Silver 클래스로 동적 대역 할당 경우 이다. 즉 Silver 클래스에 설계된 기존의 20Mbps에 추가로 15Mbps를 더 할당 하여 총 35Mbps로 서비스가 가능하여

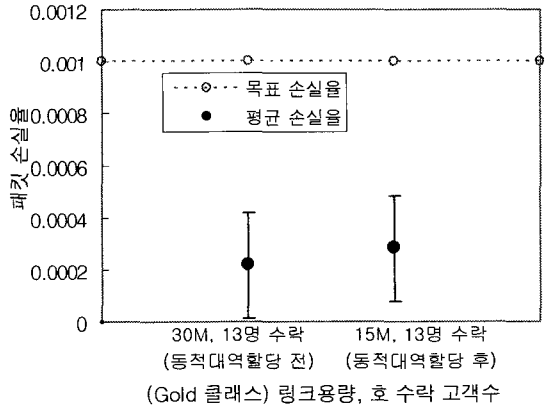


(a)

기존 30명의 서비스 고객에 추가로 22명의 Silver 클래스 고객에게 서비스 제공이 가능 하여 Silver 클래스의 호 수락 효율성이 73.33% 증가하였고 동적할당을 하기 이전에는 전체 망 사용율이 43% 였지만 동적 할당 후에는 망 사용율이 61%로 망의 이용효율 역시 증가하였음을 확인할 수 있다.

그림 12는 $C_{min}^k = 0$ 즉, 최소 대역폭을 고려하지 않고 Gold 클래스의 여분의 대역폭을 필요로 하는 클래스가 있다면 가능한 한 모두 할당하는 경우의 실험 결과이다. 따라서 Gold 클래스의 여분의 대역 17Mbps를 모두 Silver 클래스에 할당한다. 그 결과 25명을 추가로 호 수락 하여 Silver 클래스에서 호 수락을 요구하던 가입자 55명 모두가 서비스를 받을 수 있음을 알 수 있다. 하지만 이러한 방법은 상대적으로 우선 순위 에 있는 클래스에 호 거부가 증가할 수 있으므로 바람직하지 못한 결과를 초래할 수 있다.

그림 13은 그림 10에서의 최소 대역폭을 $C_{min}^k = C_{dm}^k / 2$ 로 설정하여 대역을 동적할당한 후의 패킷 손실율을 측정 한 결과이다. 평균 패킷 손실율은 20회 반복 실시하여 구하였다. 그림 13-(a)는 20Mbps로 디멘전닝 된 Silver 클래스 링크에 30명의 고객을 수락하였을 경우 패킷 손실율과 대역을 동적 할당 받아 35Mbps의 링크 용량에 52명의 고객을 수락한 경우의 패킷 손실율은 측정 한 결과 이다. 그림 13-(b)는 Gold 클래스에서 30Mbps로 디멘전닝 된 링크에서 13명을 서비스 하는 경우와 그림 13-(a)의 Silver 클래스로 Gold 클래스의 대역을 동적 할당 하여 15Mbps의 링크에서 13명에게 서비스 하는 경우



(b)

그림 13. 동적 대역 할당 후의 패킷 손실율 측정

의 패킷 손실율을 측정한 결과이다. 실험 결과 Gold와 Silver 클래스 모두 동적 대역할당을 한 후에, 패킷 손실율이 평균 0.0001정도 증가하였지만 QoS를 제공하기 위한 목표 손실율인 0.001보다 낮은 수준의 패킷 손실이 관측 되었으며 더 많은 수의 호를 수락할 수 있음을 알 수 있다.

이상의 동적 대역 할당을 수행을 위한 최소 대역폭 설정의 다양한 경우에 대하여 실험 결과 최소 대역폭을 고려한 동적 대역 할당은 보다 많은 가입자에게 서비스를 제공 할 수 있고 링크의 이용 효율 또한 높아짐을 알 수 있었다. 그러나 최소 대역폭의 범위는 계속 연구되어야 할 과제 이다.

6. 결 론

본 논문에서는 DiffServ 네트워크에서 플로우 기반의 호 수락 제어를 수행할 수 있는 메커니즘을 제안하였다. 기존의 클래스 기반의 호 관리는 실제로 클래스 내에 플로우 별로 제공되는 QoS가 엄격하게 지켜지지 못하고 클래스간 자원의 효율적 분배에 어려움이 있었지만 본 논문에서는 DiffServ 네트워크에서 플로우 기반의 호 수락 제어를 수행하되 호 수락 제어 후에는 클래스 별 패킷 처리를 할 수 있는 새로운 메커니즘을 제안하였다. 클래스의 우선 순위 에 따라 호 수락 제어의 수행 여부를 구분하였으며 사용자가 트래픽 기술자를 선언하는 경우와 선언하지 않는 경우로 구분하여 사용자의 접속 방법을 다양화 하였다. 클래스 기반의 중합된 기존의 트래픽과 플로우 기반의 신규 트래픽을 각각 계산하여서 중합할 수 있는 Gaussian 소요 대역 계산 방법을 이용하였으며, 이를 위해 필요한 클래스별 트래픽의 {평균, 분산} 도출 방법을 제시하였다. 또한 플로우 별 호 수락 제어 후에 클래스 별 패킷 처리가 원활히 이루어 질 수 있도록 하고 대역의 이용 효율 증대를 위하여 동적 대역폭 할당 및 최소 대역폭을 정하는 메커니즘을 제안 하였다. 제안된 메커니즘을 바탕으로 시뮬레이션을 통한 성능 분석을 수행하여 플로우 기반의 호 수락 제어 후에도 QoS를 보장하는 클래스 별 패킷처리가 가능함을 보였다. 본 논문의 이러한 내용은 향후 DiffServ를 지원하는 네트워크 관리에 도움을 주고 트래픽의 호 수락 제어 관리에 효율적으로 적용될 수 있을 것으로 기대된다.

참 고 문 헌

- [1] D.Sreekantan., "Implementation of a Bandwidth Broker System for resource management in Differentiated Services", <http://www.ittc.ukans.edu/~kdrao/845>, 1999.
- [2] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol(RSVP) Version 1 Functional Specification", RFC 2205, September 1997.
- [3] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997.
- [4] Paul P. White, "RSVP and Integrated Services in the Internet: A Tutorial", IEEE Communications Magazine, May 1997.
- [5] Walter Weiss, "QoS with Differentiated Service", Bell Labs Technical Journal, 1998.
- [6] De Meer, H., O'Hanlon, P., Feher, G., Blefari-Melazzi, N, Tschofenig, H., Karagiannis, G., Partain, S., Rexhepi, V., and Westberg, L., "Analysis of Existing QoS Solutions, in IETF Internet Draft draft-demeer-nsis-analysis-02.txt", 2002.
- [7] D. Goderis, "A Scalable Service-Centric IP Quality of Service Architecture for Next Generation Networks," Proceeding, NOMS 2002, Florence, Italy, 2002.
- [8] Hoang, D,B, and Li, M., "Resource discovery and control mechanisms for DiffServ, Submitted to the Proceedings of IEEE ICC'03", Anchorage, Alaska, USA, 2003.
- [9] R.R.-F.liao and Campbell, "Dynamic Core Provisioning for Quantitative Differentiated Service", IEEE IWQoS2001, May 2001.
- [10] S. Floyd, "Comments on Measurement-based Admissions control for Controlled-Load Services", <http://citeseer.nj.nec.com/correct/125785>, July 1996.
- [11] S. Jamin, S. Hhenker and P. B. Danzeig, "Comparison of Measurement-based Admission Control Algorithms for Controlled Load Ser-

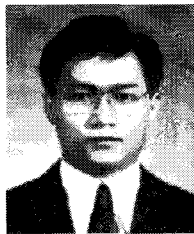
- vice”, IEEE INFOCOM97, pp.973-980, 1997.
- [12] Qiang Ren G, Ramamurthy, “A Real-time Dynamic Connection Admission Controller based on Traffic modeling, Measurement, and Fuzzy logic control”, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 2, pp. 184-196, 2000.
- [13] Roch Guerin, Hamid Ahmadi, Mahmoud Naghshineh, “Equivalent capacity and its application to bandwidth allocation in high-speed networks”, IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, pp. 968-981, 1991.
- [14] Qiang Ren G, Ramamurthy, “A real-time dynamic connection admission controller based on traffic modeling, measurement, and fuzzy logic control”, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 2, pp. 184-196, 2000.
- [15] 조병일, 유상조, 정연화, “MPLS 기반 DiffServ 망에서의 효율적인 망 dimensioning에 관한 연구”, 한국통신학회논문지, 제28권, 제5호, pp 435-447, 2003.



조 병 일

2001년 8월 인하대학교 산업공학과(공학사)
2001년 9월 ~ 현재 인하대학교 정보통신대학원 석사 과정

관심분야 : 컴퓨터 네트워크, 인터넷QoS, 트래픽 엔지니어링



유 상 조

1988년 2월 한양대학교 전자통신학과 (공학사)
1990년 2월 한국 과학 기술원 전기 및 전자공학과(공학석사)
2000년 8월 한국 과학 기술원 전자전산학과(공학박사)
1990년 3월 ~ 2001년 2월 한국통신 연구개발본부 전임 연구원

2001년 3월 ~ 현재 인하대학교 정보통신대학원 조교수
관심분야 : 인터넷QoS, 초고속 통신망 구조, 멀티미디어 네트워킹, 트래픽 엔지니어링