

차등 서비스 네트워크에 대한 성능 분석과 합성에 대한 연구

전 용 희[†] · 박 수 영^{††}

요 약

인터넷 전화, 인터넷 방송, VPN(Virtual Private Networks), 멀티미디어 서비스 등의 실시간 혹은 대역폭 요구 서비스들이 늘어남에 따라 인터넷에서도 서비스 품질(QoS: Quality of Service)에 대한 요구가 중요한 문제가 되었다. 인터넷 응용 서비스의 품질을 보장하기 위하여 통합 서비스, 차등 서비스, 그리고 MPLS(Multi-Protocol Label Switching) 등의 여러 가지 방안이 강구되고 있다. 본 논문에서는 차등 서비스(DiffServ: Differentiated Service)를 이용한 QoS 보장 기법의 성능 해석에 대하여 기술하였다. 차등 서비스의 성능이 다양한 입력 트래픽 모델과 WFQ(Weighted Fair Queueing) 가중치에 의하여 어떻게 영향을 받는지 분석하기 위하여, 랜덤, 버스티, 자기유사 입력 트래픽 모델과 다양한 입력 파라미터에 대하여 모의실험과 성능평가를 수행하였다. 성능분석 결과, 사용된 입력 트래픽 모델에 따라 패킷 지연 값과 손실 사이에 상당한 차이가 발생함을 확인할 수 있었다. 그러나, 모든 입력 트래픽 상황에서, EF(Expedited Forwarding)의 성능은 보장이 가능하고, EF와 BE(Best Effort) 클래스의 서비스 분리가 가능한 것으로 나타났다. 다음으로 성능 합성(Performance Synthesis) 문제에 대하여 기술하였다. 차등 서비스 망에 대한 보존법칙을 유도하고, WFQ 방식에서 자원의 분배(즉, 가중치)에 따른 성능 변화와 동적 행위에 대하여 분석하였다.

A Study on the Performance Analysis and Synthesis for a Differentiated Service Networks

Yong-Hee Jeon[†] · Soo-Young Park^{††}

ABSTRACT

The requirement for QoS (Quality of Service) has become an important issue as real-time or high-bandwidth services are increasing, such as Internet Telephony, Internet broadcasting, and multimedia service etc. In order to guarantee the QoS of Internet application services, several approaches are being sought including IntServ (Integrated Services), DiffServ (Differentiated Services), and MPLS (Multi-Protocol Label Switching). In this paper, we describe the performance analysis of QoS guarantee mechanism using the DiffServ. To analyze how the DiffServ performance was affected by diverse input traffic models and the weight value in WFQ (Weighted Fair Queueing), we simulated and performed performance evaluation under a random, bursty, and self-similar input traffic models and for diverse input parameters. Based on the results of performance analysis, it was confirmed that significant difference exist in packet delay and loss depending on the input traffic models used. However, it was revealed that QoS guarantee is possible to the EF (Expedited Forwarding) class and the service separation between EF and BE (Best Effort) classes may also be achieved. Next, we discussed the performance synthesis problem. We derived the conservation laws for a DiffServ networks, and analyzed the performance variation and dynamic behavior based on the resource allocation (i.e., weight value) in WFQ.

키워드: 차등 서비스(DiffServ), 성능 평가(performance evaluation), 서비스 품질(QoS: quality of service), 보존법칙(conservation law)

1. 서 론

현재의 인터넷은 모두 최선노력 서비스만 지원하고 있으나, 인터넷 전화, 인터넷 방송, VPN(Virtual Private Networks) 등의 실시간 혹은 대역폭 요구 서비스들이 늘어남에 따라 인터넷에서도 QoS(Quality of Service)가 큰 문제가 되고 있다. 인터넷에서 QoS를 제공하기 위한 방법으로, 통합서비스(IntServ: Integrated Service), 차등 서비스(DiffServ: Differentiated Service), 그리고 MPLS(Multi-Protocol Label Swicthing) 기술 등이 연구되고 있다. IntServ는 전통적인

인터넷과 QoS에 민감한 응용을 처리하기 위하여 사용되고, 신호처리 프로토콜로 RSVP(Resource Reservation Protocol)를 사용한다. 그러나, IntServ는 라우터가 개별 흐름에 대한 상태 정보를 유지해야만 하기 때문에 확장성(scalability) 문제가 존재한다. 이런 확장성 문제를 해결하기 위하여 DiffServ 모델이 제안되었고, 인터넷에 QoS를 제공하기 위한 하나의 방안으로 표준화가 진행되고 있다. 차등 서비스에서는 흐름별로 트래픽을 관리하지 않고 패킷을 클래스별(집합)로 분류해서 처리하기 때문에 RSVP와 같은 확장성의 문제가 없다[1-4]. MPLS는 입구노드에서 레이블링 작업을 통하여 IP 주소를 사용하지 않고 고속 하드웨어 스위칭을 하기 위한 기술이다[5].

본 논문에서는 DiffServ 기반 망에서 QoS 제공 기법의

† 정 희 원: 대구가톨릭대학교 공과대학 컴퓨터·정보통신공학부 교수

†† 준 희 원: 위덕대학교 공과대학 컴퓨터공학과 교수

논문접수: 2001년 9월 25일, 심사완료: 2001년 11월 16일

성능을 분석하고자 한다. DiffServ(DS) 구조는 네트워크에 진입하는 트래픽을 망의 경계에서 분류하고 조절하며, 다른 행동 집합(BA: Behavior Aggregate)에 할당되는 간단한 모델을 기초로 한다. 각 BA는 단일 코드포인트(DSCP: DS Code Point)로 확인된 후, 패킷은 네트워크 코어에서 DS 코드포인트와 관련 있는 PHB(Per-Hop Behavior)에 따라 전달된다.

차등 서비스 영역은 공통의 서비스 제공 정책과 각 노드에서 구현된 PHB 그룹의 집합으로 동작되는 DS 노드의 인접된 집합을 의미한다. DS 영역은 DS 경계 노드들로 이루어진 잘 정의된 경계를 가지며, 여기에서 그 영역을 통과하는 패킷들이 그 영역 내에서 지원되는 PHB 그룹 중 한 개의 PHB를 선택하기 위하여 적절히 마크되도록 입력 트래픽을 분류하고 조절할 수도 있다. DS 영역 내의 노드는 DS 코드포인트를 기초로 패킷에 대한 전달 행동을 선택한다. DS 경계 노드와 내부 노드 둘 다 DSCP에 기초하여 패킷에 적절한 PHB를 적용할 수 있어야 한다. 이와 같이, DS 경계 노드는 복잡한 분류와 DS 코드포인트를 기초로 패킷에 대하여 적합한 PHB를 적용하는 트래픽 조절 기능이 구현되도록 하고, 반면 내부 노드는 PHB에 해당하는 적절한 행동으로 트래픽 전달 처리를 수행한다.

PHB는 어떤 특정한 DS 행위 집합(BA)에 적용되는 DS 노드의 외부적으로 관찰할 수 있는 전달 행위(FB: Forwarding Behavior)의 묘사이다[6]. PHB는 노드가 BA에 대하여 자원을 할당하는 수단이다. PHB는 다른 PHB에 비해 그들의 자원(예, 버퍼, 대역폭) 우선 순위 혹은 상대적인 관찰 가능한 트래픽 특성(예, 지연, 손실)에 의하여 명시될 수 있다. PHB는 어떤 버퍼 관리와 패킷 스케줄링 메커니즘에 의해 노드 내에서 구현된다. PHB는 서비스 공급 정책에 관련된 행위 특성에 의하여 정의되며, 특정한 구현 메커니즘에 의한 것은 아니다. 일반적으로, 다양한 구현 메커니즘이 하나의 특정한 PHB 그룹 구현에 적합할 수도 있다. 게다가, 하나의 노드 상에 한 개 이상의 PHB 그룹이 구현되어 영역 내에서 사용될 수도 있다.

DSCP에는 Default PHB, EF PHB, AF(Assured Forwarding) PHB 등이 있다[7-9]. 특정 서비스를 요구하지 않는 패킷에 대해서는 DE(Default)로 서비스 해준다. 인터넷 라우터에서 취해지는 패킷 전달 방식인 최선 노력과 같은 정도의 서비스를 나타내며, 지연이나 손실 같은 보장된 전달 행동이 없는 IP 네트워크 상의 서비스로 볼 수 있다. EF는 라우팅 정보 갱신과 실시간 트래픽 전달에 사용되는 우선 순위가 높은 전달 방식이다. 패킷은 작은 지연과 지연 변이, 적은 손실을 가지며 대역폭 보장 서비스를 받을 수 있다. 양질의 서비스를 요구하는 트래픽을 서비스하기에 적절한 PHB이다. EF 트래픽 플로우를 엄격한 PQ(Priority Queueing)나 높은 가중치를 할당받은 WFQ(Weighted Fair Queueing) 등의 스케줄링 기법을 통해 엄격한 서비스를 제공받는다. 이러한 특성으로 인해 EF PHB에 속하는 트래픽은

가상 전용선(VLL: Virtual Leased Line)과 같은 서비스를 받을 수 있다. 그러므로 최대 비트율이나 고정된(constant) 비트율을 가진 실시간 어플리케이션을 위해 매우 좋다. EF를 위한 어플리케이션 예로는 Voice over IP, Video over IP 등이 있다. AF PHB에 속하는 트래픽은 망의 혼잡 상황에서도 트래픽의 최소 성능 속도를 보장하는 PHB이다. 본 논문에서는, 단순하게 하기 위하여 EF와 DE에 대하여만 다룬다.

본 논문에서는 네트워크 성능분석을 위한 도구인 OPNET을 사용해 차등 서비스 성능분석을 할 수 있는 시뮬레이터를 구현하고, 그 시뮬레이터로 다양한 트래픽 모델에 대한 서비스 품질 보증 기능을 시뮬레이션하고 분석한다. 그리고, 트래픽 스케줄링의 이론적 배경이 되는 보존 법칙(Conservation Law)에 대하여 기술하고, DiffServ를 위한 보존 법칙을 유도한다. 이 법칙을 기반으로 WFQ에서 자원분배(즉, 가중치)에 따른 동적 행위를 분석하고자 한다.

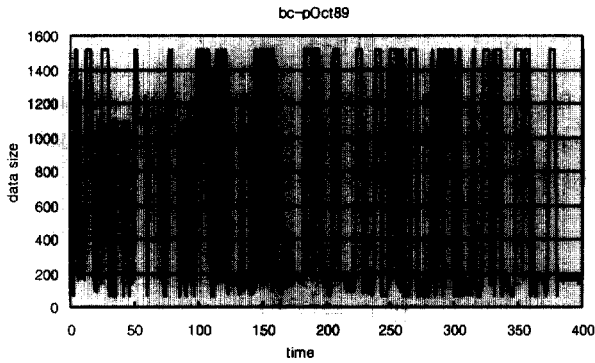
본 논문의 구성은 다음과 같다. 먼저 2장에서는 인터넷 트래픽 분석과 모델링에 대한 내용을 기술한다. 3장에서는 구현될 차등 서비스 메커니즘에 대해서 기술하고, 4장에서는 시뮬레이터 설계와 성능분석 결과를 제시하고 분석한다. 5장에서는 차등 서비스 메커니즘에 대한 보존 법칙을 유도하고, 이를 기반으로 자원 할당에 대한 동적 행위(Dynamic Behavior)에 대하여 분석한다. 마지막으로 6장에서는 결론과 향후 연구 계획으로 끝을 맺는다.

2. 트래픽 분석 및 모델링

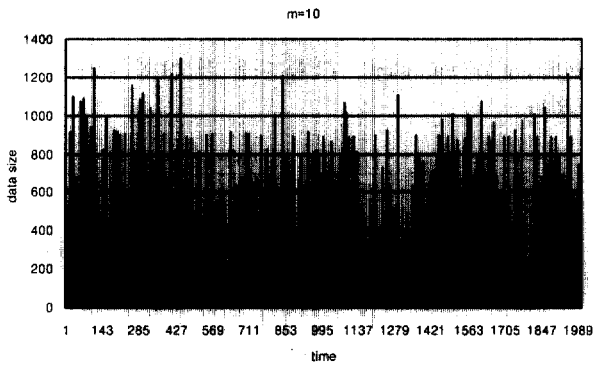
현재 제공되는 많은 인터넷 서비스 트래픽들의 특성은 기존 트래픽 성질과는 완전히 다른 자기유사성이라는 성질을 가진다는 사실이 밝혀지면서 이에 대한 연구가 진행되고 있다. 1990년대 초반에 인터넷 트래픽이 장기간 의존성(LRD: Long-Range Dependency)을 가진다는 것이 발견된 이후, 많은 MAN 트래픽, WAN 트래픽, VBR 비디오 트래픽, WWW 트래픽 등도 LRD 성질을 가진다는 것이 증명되었다[10-13].

2.1 트래픽 분석

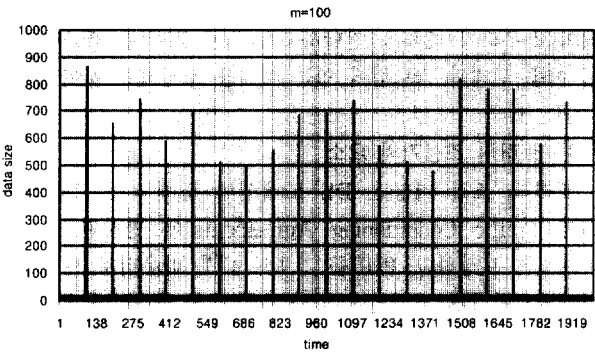
인터넷의 실측 트래픽의 자기 유사적인 특성을 확인하기 위하여 실측 트래픽 데이터를 이용하여 자기 유사성 특성을 조사하였다. (그림 1)은 실측 트래픽 데이터의 크기로 BC-pOct89 자료의 표본들을 보여준다[14]. (a)는 자료를 별개로 모두 나열한 경우의 결과이고 (b)는 10개씩 순서대로 집단화한 것이며 (c)는 100개씩 집단화한 경우의 결과를 표현한다. 직관적으로도 이들의 특성이 자기유사적인 특성을 표현한다는 것을 알 수 있다. 이것을 R/S plot과 variance-time plot을 통해서 H 값을 구해보면 $m=100$ 인 경우 H (Hurst Parameter)는 0.841794 정도로 높은 자기유사성을 가지고 있음을 확인할 수 있었다[15].



(a) m = 1인 경우의 시간 대 데이터 크기



(b) m = 10인 경우의 시간 대 데이터 크기



(c) m = 100인 경우의 시간 대 데이터 크기

(그림 1) BC-pOct89 트레이스의 자기유사성

2.2 트래픽 모델링

2.2.1 자기유사 트래픽 모델

자기 유사성은 다음과 같이 정의된다: 어떤 프로세스 $x(t)$ 가 있을 때, 임의의 실수 $a > 0$ 에 대하여 프로세스 $a^{-H}x(at)$ 가 $x(t)$ 와 같은 통계적 특성을 가진다면 $x(t)$ 는 파라미터 H ($0.5 \leq H \leq 1$)를 가지고 통계적으로 자기유사적이라 한다. H (Hurst Parameter)는 자기유사성의 정도를 나타내는 값으로, $0 < H < 1$ 사이에 존재한다. H 값이 1에 근접할수록 자기 유사성이 강함을 의미한다. 자기 유사 프로세스는 평균, 분산 및 자기 상관관이 식 (1)~식 (3)과 같이 정의된다.

$$\text{평균: } E[x(t)] = \frac{E[x(at)]}{a^H} \quad (1)$$

$$\text{분산: } Var[x(t)] = \frac{Var[x(at)]}{a^{2H}} \quad (2)$$

$$\text{자기상관: } R_x(t, s) = \frac{R_x(at, as)}{a^{2H}} \quad (3)$$

자기유사적 프로세스의 중요한 특성으로 장기간 의존성, heavy-tailed 분포, 스펙트럼 밀도 등이 있다.

- 장기간 의존성: 한 정상 프로세스가 있을 때, 이 프로세스의 자기상관 함수 $r(k)$ 가 더할 수 없는 경우(즉, $\sum_k r(k) = \infty$), 이러한 프로세스를 장기간 의존적이라 한다. 대표적인 단기간 의존적(SRD: short-range dependency) 프로세스인 포와손 모델은 자기공분산이 지수적으로 빨리 감소하는 조건을 만족하는데 비하여, LRD 프로세스는 hyperbolic하게 감소하는 자기공분산을 가지는 특성을 가지고 있다. 이런 특성이 네트워크의 성능분석에 심대한 영향을 미칠 수 있다.
- Heavy-tailed 분포: 패킷 도착간 시간이나 버스트 길이와 같은 트래픽 프로세스를 나타내는 확률밀도 함수를 특성화하기 위하여 사용되며, 확률 변수 X 의 분포가 다음 식 (4)을 만족하면 긴 꼬리라고 한다.

$$1 - F(x) = \Pr[X > x] \sim \frac{1}{x^a}, \quad (x \rightarrow \infty, a > 0) \quad (4)$$

일반적으로, 긴 꼬리 분포를 가진 확률변수는 크거나 무한인 분산값을 가진다.

- 스펙트럼 밀도: 주파수 영역에서 기술하면, LRD 프로세스의 전력 스펙트럼 밀도는 식 (5)와 같은 원래의 신호에 가까운 전력 법칙을 따른다.

$$S(w) \sim \frac{1}{|w|^r}, \quad (w \rightarrow 0, 0 < r < 1) \quad (5)$$

자기유사적인 트래픽을 생성하는 방법으로는 긴 꼬리 On/Off 구간들을 가지는 단순 On/Off 프로세스들을 다중 중첩시켜서 자기유사적 프로세스를 구축하는 방법, M/G/∞ 큐잉모델, RMD(Random Midpoint Displacement), Wavelet 계수 계산법, Hosking 방법, FFT(Fast Fourier Transform) 방법 등이 있다[10-13]. 이처럼 자기유사 트래픽을 모델링하는 기법에는 여러 가지가 있는데, 본 논문에서는 On/Off 프로세스 기법을 이용하여 구현하였다.

웹을 포함하여 현재 인터넷에서 제공되어지는 서비스를 위한 파일 구조는 heavy-tailed 분포를 가지며, 이러한 특성은 합성된 네트워크 트래픽의 자기유사성을 일으키는 원인을 제공한다. 따라서, 간단한 On/Off 프로세스를 중첩시키면 자기유사적인 결과가 나타난다. 이 때 중첩되는 스트림의 수가 증가될수록 정확한 자기유사 모델의 표현이 가능한데, 최소 150 개 이상의 긴 꼬리 분포를 가지는 스트림의 중첩이 필요한 것으로 알려져 있다. 이러한 모델의 개별

적 On/Off 소스는 데이터 통신망 터미널 모델링에 상당히 유용하다. 이 모델에서의 트래픽 소스는 On 구간동안 R의 율로 데이터를 전송하거나 Off 구간동안 비활성 상태로 남아 있게 된다. On 상태동안 보내는 시간 t_{on} 과 Off 상태동안 보내는 시간 t_{off} 는 긴 꼬리 분포를 가진다. 긴 꼬리 분포를 모델링하기 위해서는 Pareto 분포가 주로 사용된다.

2.2.2 비자기유사 트래픽 모델

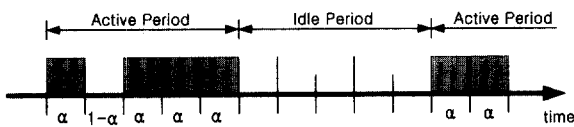
인터넷에서의 트래픽을 모델링하기 위해 사용되는 방법 중 자기유사성을 고려하지 않는 모델로는 MMPP (Markov-Modulated Poisson Process)/MMBP(Markov-Modulated Bernoulli Process), IPP(Interrupted Poisson Process)/IBP(Interrupted Bernoulli Process), Fluid Flow 모델 등이 있다. 본 논문에서는 자기 유사 트래픽 모델과 비교를 위하여 비자기유사 트래픽 모델로 랜덤 트래픽 및 IBP 트래픽 모델을 사용하였다. 랜덤 트래픽 모델을 포와손 프로세스의 이산버전인 베르누이 프로세스에 의해 표현한다. 베르누이 프로세스에서 각 슬롯에서의 도착확률은 각 슬롯 사이가 독립적으로 p이다. (그림 2)는 랜덤 트래픽 모델에서의 시간 슬롯을 보여주고 있다.



(그림 2) 랜덤 트래픽 모델에서의 시간 슬롯

지수분포를 가지는 On(active period) 상태와 또 하나의 다른 독립적인 지수분포를 가지는 Off(silent period) 상태가 교대로 나타나는 프로세스인 IPP모델은 On-Off 트래픽의 대표적인 모델이다. IPP의 이산 모델로 IBP가 있다. IBP 모델에서의 시간은 슬롯화 되어 있으며 그 크기는 매체에서의 하나의 패킷 시간과 동일한 것으로 가정한다(그림 3 참조).

프로세스가 활동 상태에 있을 때 다음 슬롯에서 확률 p를 가지고 그 상태에 머물러 있거나 확률 1-p를 가지고 휴지 상태로 이동한다. 만약 프로세스가 휴지 상태에 있다면 확률 q를 가지고 휴지 상태에 계속 머물고 확률 1-q만큼 활동 상태로 변할 것이다. 일반적으로 프로세스가 활동기간 내에 있다면 각 슬롯은 확률 α 만큼 패킷을 포함할 것이다. 본 논문에서는 α 값을 1로 설정해, 활동 상태에 있는 모든 슬롯은 패킷을 가진다고 가정하였다[16].



(그림 3) 버스티 트래픽 모델에서의 시간 슬롯

IBP 프로세스의 전이 확률(transition probability) 행렬은 다음 식 (6)과 같다.

$$P = \begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix} \tag{6}$$

Π_A, Π_I 를 각각 활동 및 휴지 상태의 정상-상태(steady-state) 확률로 정의하면, 정상 상태 방정식 $\Pi P = \Pi$ 로부터, 다음과 같은 식을 구할 수 있다.

$$\Pi_A = \frac{1-q}{2-p-q} \tag{7}$$

$$\Pi_I = \frac{1-p}{2-p-q} \tag{8}$$

여기서, Π_A 는 평균 도착율을 의미한다.

d를 연속적인 패킷들 사이의 도착간 시간(inter-arrival time)으로 정의하면, 도착간 시간의 제곱 변화 계수(squared coefficient of variation), C^2 을 다음 식 (9)에 의하여 구할 수 있다.

$$C^2 = \frac{Var(d)}{[E(d)]^2} = \frac{(p+q)(1-p)}{(2-(p+q))^2} \tag{9}$$

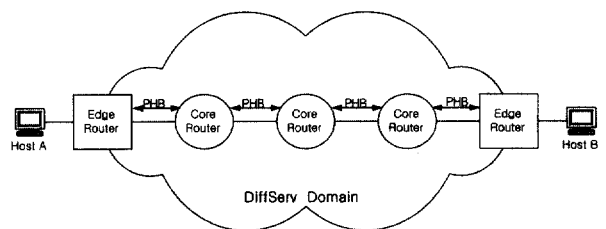
본 논문에서는, 매개변수 C^2 을 균집성(burstiness)의 척도로 사용한다.

3. 차등 서비스 메커니즘

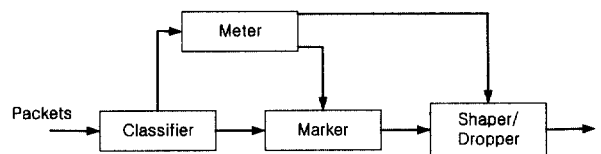
차별 QoS를 제공할 수 있는 DiffServ의 전제는 네트워크 내의 라우터가 패킷을 전달하도록 다른 PHB를 제공함으로써 이중 트래픽 스트림에 있는 패킷을 조절하는 것이다.

3.1 트래픽 조절기의 요소

트래픽 조절 기능은 DS 도메인에 입력되는 트래픽이 TCA(Traffic Conditioning Agreements)에 명시된 규칙을 따르도록 보장하고, PHB를 기초로 내부 라우터로 트래픽을 전달하도록 DS 도메인에 있는 DS 에지 라우터에 의해 수행된다. (그림 4)는 DiffServ 단일 도메인을 보여주며, (그림 5)는 트래픽 조절기의 블록 다이어그램을 보여준다.



(그림 4) DiffServ 단일 도메인



(그림 5) 트래픽 조절기 블록 다이어그램

(그림 5)에서 보여주는 트래픽 조절기 요소는 트래픽 분류기(Classifier), 미터(Meter), 마커(Marker), 셰이퍼(Shaper)로 이루어져 있으며, 반드시 4가지 모든 요소를 포함할 필요는 없다. 각 요소의 기능은 다음과 같다[17, 18] :

- 분류기 : 패킷 헤더의 내용을 기초로 트래픽 스트림에 있는 패킷을 선택한다. BA 분류와 MF(Multi-Field) 분류로 나눈다. BA 분류는 DS 필드만을 기초로 하고, MF 분류는 근원지/목적지 주소, DS 필드, 프로토콜 ID, 근원지/목적지 포트와 같은 전송 계층 헤더 필드 등의 패킷 헤더에 있는 여러 필드의 조합으로 분류를 한다.
- 미터 : 미터는 분류기에 의해 선택되어진 트래픽 스트림상의 패킷 도착율을 확인하여 TCA의 트래픽 프로파일에 일치하는지를 결정해 in-profile/out-of-profile을 수행한다.
- 마커 : 패킷은 분류된 후에 DE, AF, EF 클래스로 마킹된다. 리마킹은 AF PHB 패킷이 계약된 트래픽을 제한을 위반할 때 발생하고 out-of-profile이 된다. 이런 패킷은 DE 패킷으로 리마킹 된다.
- 드롭퍼/셰이퍼 : 만약 EF 패킷이 out-of-profile이라면 드롭퍼에 의해 즉시 탈락된다. 셰이핑은 지터를 제거하기 위해 에지 노드나 경계 노드에서 발생한다.

3.2 큐잉 정책(Queueing Policy)

본 논문에서는 차등 서비스의 구현을 위하여 WFQ 스케줄링 정책을 사용하였다. 그리고 WFQ와의 비교 목적으로 Round Robin 방식을 사용한다.

- ① Round Robin : RR 디큐잉 알고리즘은 큐 안의 패킷을 디큐하고, 다음 큐로 바뀌어 다음 패킷에 의해 차례로 디큐된다. 이 큐에서 가용 패킷이 없을 때, 모든 큐는 검사 받게 되고 존재하는 패킷은 디큐된다. 이런 알고리즘은 간단하지만 차등 서비스 구조에 적합하지 못하다. 예를 들면, 두 개의 트래픽 클래스, EF와 BE 서비스가 있다고 하자. EF와 BE 패킷이 큐 안에 있을 때, 패킷은 차례로 디큐되고 긴급하게 디큐되는 것은 없다. 이런 경우 차등 서비스 구조의 의도가 아닌 50% EF와 50% BE의 대역폭 공유가 준수된다.
- ② WFQ(Weighted Fair Queueing) : 각 트래픽 흐름에 가중치를 할당하여 차등 서비스를 제공할 수 있는 알고리즘으로 독립적인 큐를 가지며 폭주시에도 모든 트래픽 흐름이 일정 부분의 대역폭을 할당받을 수 있다.

본 논문에서는, OPNET을 이용하여 차별 서비스를 지원할 수 있는 DiffServ 네트워크의 핵심 부분인 DiffServ 에지 라우터 구조를 설계하였다. 에지 라우터의 구성 요소는 TCB, 버퍼 관리, 스케줄러로 이루어져 있다. TCB는 트래픽 분류기능과 조절기능이 조합되어 있는 블록이다. 트래픽 분류기는 BA 분류기를 사용하여, DSCP 필드를 이용하여 패

킷을 분류하였다. 마킹된 패킷은 큐잉 단계(큐잉/디큐잉)로 넘어가는데, 본 논문에서는 큐잉 요소 메커니즘으로 WFQ, WRR(Weighted Round-Robin)과 같은 패킷 스케줄링 알고리즘을 사용해 차별 서비스를 지원하였다.

4. 시뮬레이터 설계 및 성능분석

4.1 DiffServ 시뮬레이터 설계 및 구현

이 장에서는 소프트웨어 도구 OPNET을 사용해, DiffServ 성능을 모의실험 할 시뮬레이터를 설계하고 구현하였다. 패킷 크기는 512 바이트이며, 패킷 전송율은 155(Mbps)로 설정했다. DiffServ 네트워크가 올바르게 동작하는지 검사하기 위해서 평균 종단간 지연 통계를 이용하였는데, 이 통계치는 시스템의 성능을 분석하고 개선을 연구하기 위해 사용되었다. 먼저 DiffServ 시뮬레이터에서 구현된 트래픽 조절 기능들을 설명한다.

4.1.1 트래픽 조절 기능 구현

- 분류(Classifying) : BA 분류는 코드포인트만으로 분류하여 구현되었다. 패킷간 차별화를 만들기 위해 IPv4 헤더에 있는 DSCP에 접근하도록 만들었다.
- 마킹(Marking) : 도착하는 패킷은 코드포인트의 클래스 선택자에 의해 트래픽 클래스 EF, BE 트래픽으로 분리된 후, 마킹 되고 큐잉된다. “classifier” 상태에서 입력 스트림을 통해 입력되는 패킷을 BA 분류 방법으로 분류하고, “marking” 상태에서 marker에 의해 DSCP에 마킹을 한다.
- 큐잉(Queueing) : DiffServ 구조의 장점을 얻는데 큐잉 알고리즘은 매우 중요하다. 여기서는 Round Robin과 WFQ 스케줄링 알고리즘이 시스템에서 구현되었다. 마킹된 패킷들은 해당하는 각 큐에 입력이 된다. 그 다음 WFQ 트래픽 스케줄링을 사용해, 가중치를 부여받은 패킷 스트림은 차등화된 대역을 할당받는다.

4.1.2 패킷 생성기 구현

(1) 자기 유사 트래픽 모델 : 현재 인터넷에서의 대부분 트래픽들이 자기유사성을 가진다는 연구결과가 발표되었다. 그래서 본 연구에서도 자기유사 트래픽을 적용한 DiffServ 시뮬레이터의 결과를 분석하기 위해 자기유사 트래픽 생성기를 구현해 이용하였다. 자기유사 트래픽을 생성하는 방법으로는 여러 가지가 있는데, 긴 꼬리 On/Off 구간들을 가지는 단순 On/Off 프로세스들을 다수 중첩시켜서 자기유사적 프로세스를 구축하는 방법을 사용했다. (그림 6)은 자기유사 트래픽 모델에 의해 구현된 패킷 생성기 알고리즘을 보여준다.

```

if (ON state)
{
    incrementing no. of streams in ON state ;
    scheduling next OFF state ;
    heavy-tailed distribution modeling using Pareto distribution ;
}
    
```

```

recalculating interarrival times between packets ;
if (gen == 0)
    scheduling next packet generation time ;
}
if (OFF state)
{
    decrementing no. of streams in ON state ;
    recalculating interarrival times between packets ;
    scheduling next ON state ;
    long-tailed distribution modeling using Pareto distribution ;
}
    
```

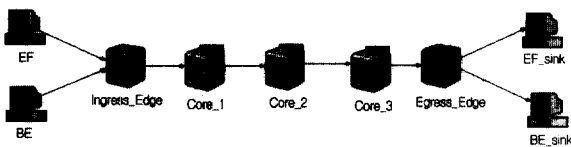
(그림 6) 자기 유사 트래픽 생성기 알고리즘

(2) 랜덤 트래픽 모델 : 랜덤 트래픽 생성기에서의 패킷 도착은 파라미터 traffic_load를 가지며 베르누이 프로세스에 의해 생성된다. 파라미터 traffic_load는 사용자 입력 파라미터이며, 그 값은 $0 \leq \text{traffic_load} \leq 1$ 이다. traffic_load는 임의의 타임 슬롯에 패킷이 생성될 확률이고 $(1 - \text{traffic_load})$ 는 패킷이 생성되지 않을 확률이다.

(3) 버스티 트래픽 모델 : 버스티 트래픽의 생성은 IBP에 의하여 수행된다. IBP 프로세스에서의 사용자 입력 파라미터는 트래픽 부하량과 버스티니스(burstiness) C^2 의 값, 그리고 활동기간 중에 있는 각 시간 슬롯마다 패킷을 포함할 확률인 α 이다. 이런 파라미터의 값을 다양하게 설정해 버스티 특성을 가진 트래픽에 대해서는 DiffServ 시뮬레이터가 어떤 성능을 나타내는지를 분석하였다.

4.2 성능분석 모델

DiffServ 기능 검증을 위하여 앞 장에서 구현된 DiffServ 시뮬레이터를 사용하여 DiffServ 네트워크를 구현해 인터넷 QoS 지원 기능에 대한 성능 평가를 수행할 필요가 있다. 그래서 (그림 7)과 같은 성능 분석 모델에 대하여 DiffServ 시뮬레이터를 이용하여 모의실험을 수행하고 성능 평가를 수행하였다. 내부 라우터 3개를 포함하여 총 5개의 라우터로 구성되어 있으며, 트래픽 발생기는 EF와 BE에 대하여 별도로 구성하였다.



(그림 7) 성능 분석 모델

4.3 성능분석 결과

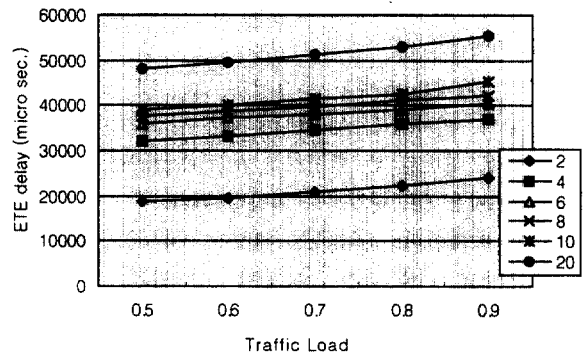
본 논문에서는 구현된 각 트래픽 모델(즉, 랜덤, 버스티, 자기 유사 모델)에 대해 EF 및 BE PHB 두 가지 트래픽 클래스로 차등 서비스에 대한 모의실험을 수행하였다.

4.3.1 랜덤 트래픽에 대한 결과 및 분석

(1) 평균 종단간 지연 비교

(그림 8)은 네트워크 용량 대비 트래픽 부하량 50%에서

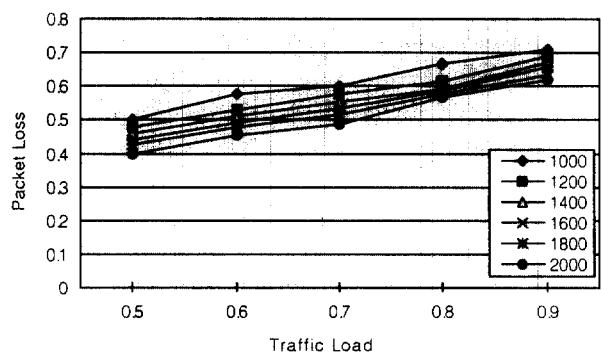
90%까지에 대하여 EF와 BE 트래픽의 평균 종단간 지연을 보여주고 있다. 예상할 수 있듯이, 가중치가 높을수록 BE 트래픽에 대한 지연이 증가하는 것을 볼 수 있다. 대조적으로, EF 트래픽의 평균 종단간 지연은 감소하였다. EF 지연 값 중 가장 큰 값인 트래픽 부하 0.9와 가중치 2에서도 200 마이크로 초 이하이기 때문에, 모든 파라미터에 대하여 무시할 수 있을 정도의 작은 값을 보여준다. 반면 BE 트래픽의 평균 종단간 지연은 증가하는 것을 알 수 있다. 즉, EF와 BE 서비스의 분리로 현재 인터넷에서는 제공되지 않는 차등화 서비스가 가능함을 보여준다. 높은 전체 인가 부하에서도 EF 서비스는 QoS가 보장되는 반면에 BE 트래픽의 종단간 지연은 상대적으로 크게 증가됨을 볼 수 있다.



(그림 8) 가중치 값에 따른 EF와 BE 클래스의 평균 종단간 지연 비교

(2) 패킷 손실

(그림 9)는 트래픽 부하량 50%에서 90%까지에 대하여 EF와 BE 트래픽의 패킷 손실을 보여주고 있다. 가중치는 5로 설정을 했고, 버퍼 크기에 따른 패킷 손실 확률을 보여준다. 버퍼 크기는 1000에서 2000까지의 범위로 설정을 했다. EF와 BE 서비스간의 패킷 손실이 확실히 차별화 되는 것을 비교할 수 있다. 높은 전체 인가 부하에서도 EF 서비스는 QoS가 보장되는 반면에 BE 트래픽의 패킷 손실은 패킷 크기가 증가함에 따라 패킷 손실이 감소되기는 하지만 EF에 비해 상대적으로 많이 발생함을 볼 수 있다. EF 트래픽의 경우는 버퍼 크기에 상관없이 전혀 패킷의 손실이



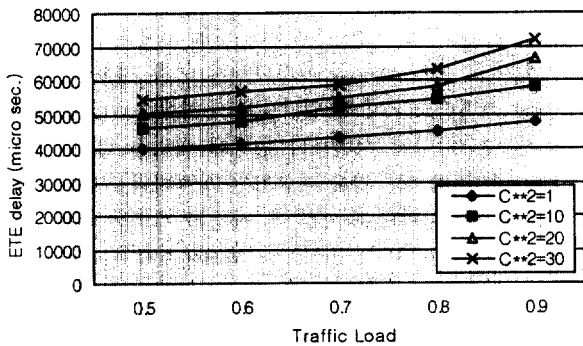
(그림 9) EF와 BE 서비스의 패킷 손실 비교

발생하지 않았다. 이는 EF 클래스에 대하여는 패킷을 우선적으로 처리하기 때문에 패킷이 누적되지 않아 손실이 발생하지 않기 때문이고, BE 클래스에 대하여는 우선 취급을 하지 않아 패킷의 누적되는 양이 증가하게 되어, 버퍼 오버플로우가 발생하여 패킷 손실이 발생하는 것으로 판단된다. 따라서, 버퍼 크기를 결정할 때 이런 점을 고려하여, BE 클래스에 대하여는 EF보다 훨씬 더 큰 버퍼를 할당해야 할 것으로 분석된다.

4.3.2 버스티 트래픽에 대한 결과 및 분석

(1) 평균 종단간 지연

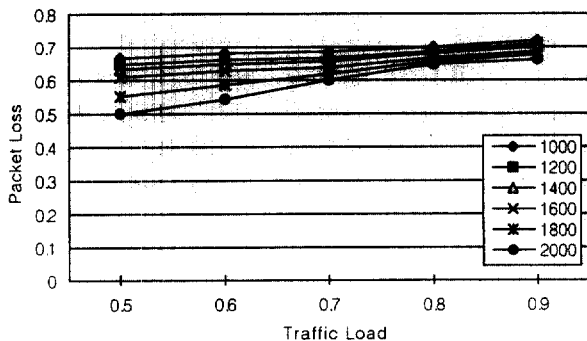
(그림 10)은 트래픽 부하량 50%에서 90%까지에 대하여 EF와 BE 트래픽의 평균 종단간 지연을 보여주고 있다. 버스티니스의 척도인 C^2 값은 1에서 30의 범위로 하고, 가중치는 5로 설정하였다. 버스티 입력 트래픽인 경우에도 EF 클래스에 대하여는 종단간 QoS가 보장됨을 보여준다. 그러나, 버스티니스의 값이 커질수록 BE 클래스의 지연은 크게 증가된다. 이것은 대역폭 보장을 받지 못하는 BE 서비스의 지연 특성이 트래픽의 버스트 성으로 악화되기 때문이다. (그림 8)에서 랜덤 트래픽의 경우 동일한 가중치 5에서, 부하 0.5에서 0.9까지의 BE 트래픽의 지연 범위는 30에서 40msec에 걸쳐 있는 반면, (그림 10)에서는 패킷 음성 트래픽의 대표적 버스티니스 값인 $C^2 = 20$ 인 경우, 지연 값이 50에서 60msec 후반까지 1.5배 이상 증가 된 것을 알 수 있다.



(그림 10) EF와 BE 클래스의 평균 종단간 지연 비교

(2) 패킷 손실

(그림 11)은 (그림 10)과 같은 가중치와 $C^2 = 20$ 에서 EF



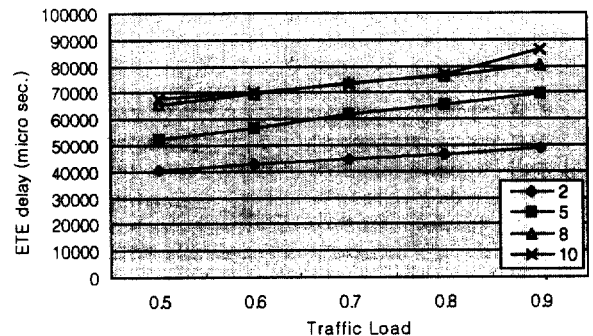
(그림 11) EF와 BE 클래스의 패킷 손실 비교

와 BE 트래픽의 패킷 손실을 보여주고 있다. 버퍼 크기에 관계없이, EF 트래픽은 패킷 손실이 전혀 발생하지 않는 반면, BE 트래픽의 패킷 손실은 랜덤 트래픽에 비하여 상대적으로 높은 것을 볼 수 있다. 특히, 입력 부하가 낮을 때 랜덤 트래픽과 버스티 트래픽 사이의 손실에서 많은 차이가 나는 것을 확인 할 수 있다. 이것은 입력 부하가 낮을 때에도, 입력 트래픽의 버스티니스가 패킷 손실에 많은 영향을 주기 때문인 것으로 분석된다.

4.3.3 자기유사 트래픽에 대한 결과 및 분석

(1) 평균 종단간 지연

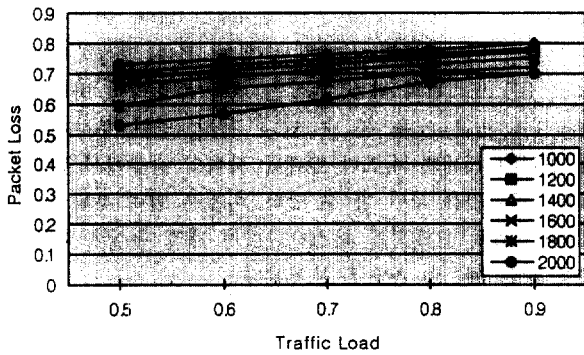
(그림 12)는 인가 부하량에 대하여 EF와 BE 클래스의 평균 종단간 지연을 보여주고 있다. 현재 인터넷망의 트래픽 특성으로 알려진 자기유사 특성을 가진 트래픽에 대해서도 DiffServ에서 WFQ 정책을 적용할 경우, EF 클래스의 종단간 지연 특성도 만족되는 것으로 분석된다. 자기유사 트래픽에 대해서도 위의 트래픽들과 마찬가지로 가중치가 높을수록 EF 트래픽의 평균 종단간 지연이 감소하는 것을 확인 할 수 있었고, BE 트래픽의 평균 종단간 지연은 증가하는 것을 볼 수 있다. 주목할 점은, 랜덤 트래픽과 버스티 트래픽의 종단간 지연보다는 EF와 BE 클래스의 지연 값 모두가 많이 높아 졌다는 사실이다. (그림 10)과 (그림 12)의 지연을 비교하여 보면, 자기 유사 트래픽으로 인한 지연은 버스티 트래픽에서 버스티니스 값 $C^2 = 20$ 에서 30 사이의 지연 값과 같은 정도의 지연을 야기 시키는 것으로 분석할 수 있다.



(그림 12) 자기유사 입력 트래픽 모델에 대한 EF와 BE 클래스의 평균 종단간 지연 비교

(2) 패킷 손실

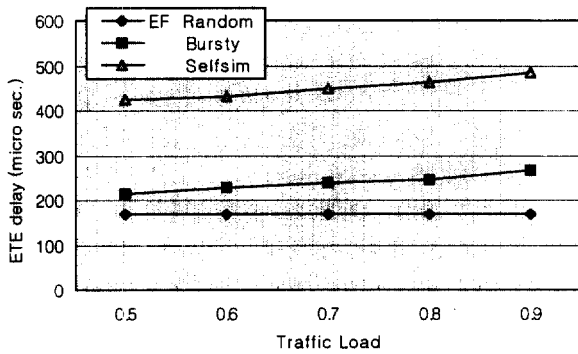
(그림 13)은 인가 부하량에 대하여 EF와 BE 클래스의 패킷 손실을 보여주고 있다. 자기유사 트래픽에 대해서도 위의 트래픽들과 마찬가지로 동일한 환경에서, BE 트래픽은 패킷 손실이 많이 발생하는 반면 EF 트래픽은 패킷 손실이 전혀 발생하지 않음을 알 수 있다. 현재 인터넷망의 트래픽 특성으로 알려진 자기유사 특성을 가진 트래픽에 대해서도 DiffServ가 적용될 경우, EF 클래스의 패킷 손실도 발생하지 않기 때문에 종단간 QoS 특성이 만족될 것으로 분석된다. 그러나 지연 특성과 마찬가지로, 랜덤 트래픽의 BE 클래스 패킷 손실 보다는 많이 높아진 것을 알 수 있다.



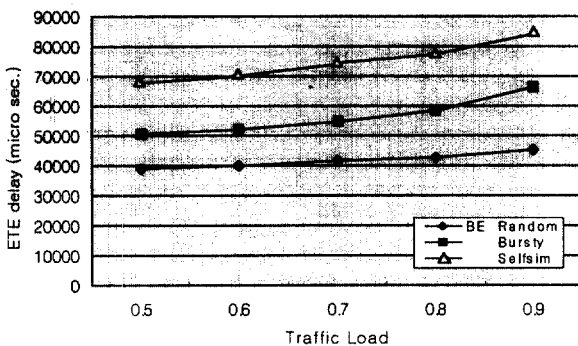
(그림 13) EF와 BE 클래스의 패킷 손실 비교

4.3.4 트래픽 모델 별 성능 비교

(그림 14)(a)와 (b)는 트래픽 모델에 따라 평균 종단간 지연 통계치로 EF와 BE 클래스의 성능을 분석한 결과를 나타내었다.



(a)

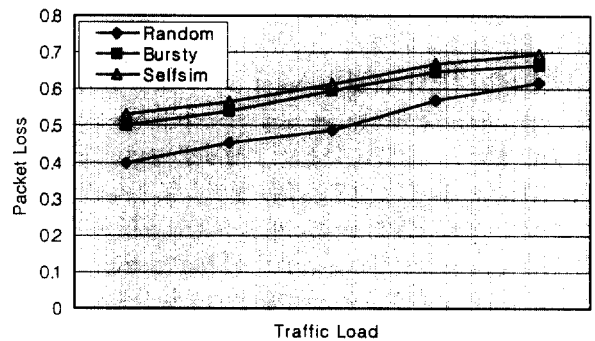


(b)

(그림 14) 트래픽 모델에 대한 평균 종단간 지연

동일한 EF 클래스이더라도 자기유사 트래픽 특성을 가진 EF 클래스가 가장 높은 종단간 지연을 보였고, 그 다음이 버스티 특성을 가진 EF 트래픽 클래스의 종단간 지연이 두 번째로 높았다. BE 클래스 트래픽도 동일한 결과로 자기유사 특성을 가진 BE 트래픽이 가장 높은 종단간 지연이 나타남을 보였다. 그러나 절대적인 지연 값의 차이는 다소 존재하나, 트래픽 특성에 상관없이 EF 트래픽의 종단간 QoS는 만족됨을 알 수 있다.

(그림 15)는 트래픽 특성에 따라 패킷 손실 통계치로 EF와 BE 클래스의 성능을 비교한 결과를 나타내었다. 트래픽 특성에 상관없이 모든 EF 트래픽은 패킷 손실 발생이 없었다. 그러나 BE 트래픽의 경우는 자기유사 트래픽, 버스티 트래픽, 랜덤 트래픽 순으로 패킷 손실 발생율이 높았다. 그러나, 에지라우터를 일단 통과한 트래픽은 트래픽이 평활화(smoothing)되어 트래픽 모델간에 별다른 특성을 찾을 수 없었다. 그러므로, 입력 소스에서 트래픽 특성을 좋게 하기 위하여 웨이핑과 같은 기능이 절대적으로 필요함을 알 수 있다.



(그림 15) 트래픽 특성에 따른 패킷 손실 비교

5. 동적 행위 분석(Dynamic Behavior Analysis)

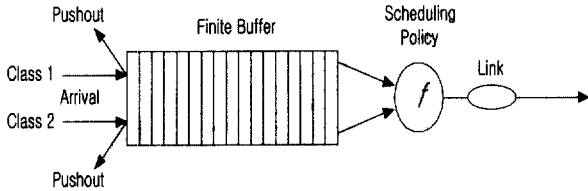
본 장에서는 차등 서비스의 EF와 BE 클래스간에 할당되는 가중치에 따른 서비스 매커니즘의 동적 행위를 분석한다. 차등 서비스는 기존의 인터넷에서 채택하고 있는 최선-노력 서비스가 아닌 서비스의 종류에 따라서 QoS를 차별화하여 서비스하는 것이 기본적인 원칙이다. 이것은 보존 법칙(Conservation Law)으로 잘 설명될 수 있다[19]. 어떤 보존 법칙도 “풀에게 갓기 위하여 피터에게서 빌린다”는 원칙을 적용함으로써 직관적으로 묘사될 수 있다. 즉, 한 개 이상의 트래픽 클래스를 가진 큐잉 시스템에서, 한 클래스의 트래픽에 주어지는 우선적인 취급은 성능 측도에 관계없이 다른 클래스에서의 비용으로 주어진다. 이는 다음과 같다.

5.1 차등 서비스 보존 법칙 유도

일반적인 성능 측정은 유한 버퍼에서 패킷 지연과 손실이 기 때문에, 일반성의 손실 없이, 본 장에서는 서비스 정책이 HOL(Head-Of-Line) 정책이고, 버퍼 공간 할당은 LIFD (Last In First Drop) 알고리즘에 따라 실행되는 것으로 가정한다. 즉, 푸시아웃(pushout)된 패킷은 가장 최근에 큐에 들어온 패킷이다. 푸시아웃된 패킷은 시스템에 일단 들어오면 지연에 기여를 하기 때문에, 푸시아웃된 패킷 지연은 지연 측정에 포함된다.

일반적인 차등 서비스 트래픽 제어를 위한 큐잉 모델이 (그림 16)에 보여준다. (그림 16)의 모델은 보존 법칙을 유도하기 위한 단순화된 일반 모델이고, 4장의 시뮬레이션에

사용된 모델과는 다르다. 이 모델에서 도착(arrival)과 출발(departure)에 대한 제어를 버퍼링 및 스케줄링 정책을 통하여 실행할 수 있다. 단순하게 하기 위하여, 본 논문에서는 2가지의 클래스에 대하여만 고려하였다. 2가지 클래스 이상에 대하여도 일관성의 상실 없이 본 장에서의 법칙이 사용될 수 있다.



(그림 16) 트래픽 제어를 위한 일반 큐잉 모델

$X_i(f, p)$ 는 푸시아웃 스킴 p (손실 정책)가 사용될 때 서비스 정책 f (즉, DiffServ에서는 WFQ 정책) 하에서 버퍼에 있는 클래스 $i(i = 1, 2)$ 패킷의 평균수로 정의한다. $ED_i(f, p)$ 는 f 와 p 하에서, 클래스 i 패킷의 응답 시간의 기대값으로 정의하고, $\epsilon_i(f, p)$ 는 f 와 p 하에서, 각 클래스 i 를 위한 안정 상태의 패킷 손실 확률이다.

f 와 p 가 작업-보존(work-conserving) 정책인 한, 정책 f 가 선택적인 패킷 폐기 기법이 아니기 때문에, 어떤 서비스 정책 f 를 가지는 각 트래픽 클래스에 대하여도 동일한 패킷 손실을 가진다. 그러므로 $\epsilon_i(f, p) = \epsilon_i(p)$ 로 적을 수 있다. 그러나, 서비스 정책은 각 클래스에 대한 다른 응답 시간 특성을 초래한다.

정리 1 : 유한 버퍼 시스템과 IID(independent and identically distributed) 도착 프로세스인 경우, 다음의 불변 관계가 성립한다 :

$$X_1(f, p) + X_2(f, p) = C \tag{10}$$

증명 : C 는 안정 상태에서 평균 전체 큐 크기이고 정책 f 와 p 와는 독립적이다. 버퍼 오버플로우나 푸시아웃 때문에, 클래스 1 혹은 클래스 2의 패킷이 손실되든 간에 관계없이, 정책 f 와 p 가 작업-보존인 한 버퍼에 있는 전체 패킷 수에는 영향을 미치지 않는다. 그러나, $X_1(f, p)$ 와 $X_2(f, p)$ 각각은 확실히 정책에 의존한다. 버퍼에 있는 전체 패킷 수는 정책 f 와 푸시아웃 스킴 p 에 독립적이다. 그러므로, 정리가 성립한다.

$\lambda_i(p)$ 를 클래스 i 패킷에 대한 유효 도착률이라 하자. 리틀의 법칙을 사용해 식 (10)으로부터, 다음의 보조 정리(Lemma)를 얻을 수 있다. $ED_i(f, p)(i = 1, 2)$ 는 정책 f 와 p 아래서 클래스 i 에 대한 평균 지연을 의미한다.

보조 정리 1 : 유한 버퍼 시스템과 IID 도착 프로세스인 경우, 다음의 불변 관계가 성립한다 :

$$\lambda_1(p)ED_1(f, p) + \lambda_2(p)ED_2(f, p) = C \tag{11}$$

C 는 f 와 p 에 독립적인 상수이다.

만약 패킷 손실 확률이 0(zero)에 접근한다면, 식 (11)에서 $ED_i(f, p)$ 는 무한 버퍼 시스템에서 성취되는 지연과 매우 근접하다. 다른 한편, 작업-보존 정책 하에서, $\lambda_i(p)$ 는 0으로 가는 것을 허용하지 않는다, 그래서 $ED_i(f, p)$ 역시 임의로 0으로 근접할 수 없다.

$\Pr(1 > 2)$, $\Pr(2 > 1)$, $B_1(p)$, $B_2(p)$ 를 다음과 같이 정의한다.

- ① $\Pr(1 > 2)$ 는 정책 p 가 클래스 1에게 더 높은 우선 순위를 주는 시간의 부분(즉, 확률).
- ② $\Pr(2 > 1)$ 는 정책 p 가 클래스 2에게 더 높은 우선 순위를 주는 시간의 부분(즉, 확률).
- ③ $B_1(p)$ 는 클래스 1의 패킷이 클래스 2의 패킷에 의하여 푸시아웃되는 확률.
- ④ $B_2(p)$ 는 클래스 2의 패킷이 클래스 1의 패킷에 의하여 푸시아웃되는 확률.

그러면, 식 (10)은 식 (12)와 같이 다시 쓸 수 있다 :

$$\begin{aligned} X_1(f, p) + X_2(f, p) = & [(1 - B_1(p)) \cdot X_1(f, p) + B_1(p) \cdot X_1(f, p) + X_2(f, p)] \cdot \Pr(2 > 1) \\ & + [X_1(f, p) + (1 - B_2(p)) \cdot X_2(f, p) \\ & + B_2(p) \cdot X_2(f, p)] \cdot \Pr(1 > 2) = C \end{aligned} \tag{12}$$

식 (12)에서, 클래스 2가 높은 우선권을 가질 시간동안, 첫 번째 항 $(1 - B_1(p)) \cdot X_1(f, p)$ 는 성공적으로 전송되는 평균 클래스 1의 가상 부하(virtual load)를 표시한다. 두 번째 항 $B_1(p) \cdot X_1(f, p)$ 는 클래스 1에 대한 평균 푸시아웃된 가상 부하를 나타낸다. 세 번째 항 $X_2(f, p)$ 는 성공적으로 전송된 클래스 2의 평균 가상 부하를 나타낸다. 클래스 1이 더 높은 우선권을 가질 동안, 네 번째 항 $X_1(f, p)$ 은 성공적으로 전송되는 클래스 1의 평균 가상 부하를 표시한다. 다섯 번째 항 $(1 - B_2(p)) \cdot X_2(f, p)$ 는 성공적으로 전송된 클래스 2의 평균 가상 부하를 표시한다. 마지막 항목 $B_2(p) \cdot X_2(f, p)$ 는 푸시아웃된 평균 클래스 2 가상 부하를 나타낸다.

B_i 는 도착하는 클래스 i 의 패킷이 버퍼 충만을 발견하는 확률을 나타내고, $\overline{B_i(p)}$ 는 클래스 i 가 높은 우선권을 가질 동안, 두 가지 트래픽 클래스에 대하여, 클래스 i ($i \neq j$) 패킷이 클래스 j 를 푸시아웃하는 확률을 표시한다. 그러므로, $B_i(p) = \overline{B_j(p)}$ ($i \neq j$)이다. 식 (11)에 있는 $\lambda_i(p)$ 는 다음과 같이 적을 수 있다 :

$$\begin{aligned} \lambda_1(p) = & \lambda_1 \{ [(1 - B_1) + \overline{B_1(p)}] \cdot \Pr(1 > 2) \\ & + [(1 - B_1)(1 - B_1(p)) + B_1(p)] \cdot \Pr(2 > 1) \} \\ = & \lambda_1 \{ [1 - B_1 + \overline{B_1(p)}] \cdot \Pr(1 > 2) \\ & + [(1 - B_1)(1 - B_1(p)) + (1 - B_1)B_1(p)] \cdot \Pr(2 > 1) \} \\ \lambda_2(p) = & \lambda_2 \{ [(1 - B_2)(1 - B_2(p)) + B_2(p)] \cdot \Pr(1 > 2) \\ & + [(1 - B_2) + \overline{B_2(p)}] \cdot \Pr(2 > 1) \} \end{aligned}$$

$$= \lambda_2 \{ [(1 - B_2)(1 - B_2(p)) + (1 - B_2)B_2(p)] \cdot \Pr(1 > 2) + [(1 - B_2) + B_2 B_2(p)] \cdot \Pr(2 > 1) \} \quad (13)$$

각 클래스에 대한 평균 유효 도착율(effective arrival rate)을 다음과 같이 정의한다 :

$$\begin{aligned} \lambda_{11}(eff) &\triangleq \lambda_1 [(1 - B_1) + B_1 B_1(p)] \\ \lambda_{12}(eff) &\triangleq \lambda_1 (1 - B_1)(1 - B_1(p)) \\ \lambda_{13}(pushed) &\triangleq \lambda_1 (1 - B_1) B_1(p) \\ \lambda_{21}(eff) &\triangleq \lambda_2 (1 - B_2)(1 - B_2(p)) \\ \lambda_{22}(pushed) &\triangleq \lambda_2 (1 - B_2) B_2(p) \\ \lambda_{23}(eff) &\triangleq \lambda_2 [(1 - B_2) + B_2 B_2(p)] \end{aligned} \quad (14)$$

위의 식에서, $\lambda_{11}(eff)$ 는 클래스 1이 높은 우선권을 가질 시간 동안, 성공적으로 전송된 패킷에 대한 평균 클래스 1 도착율을 나타낸다. $\lambda_{12}(eff)$ 는 클래스 2가 높은 우선권을 가질 시간 동안, 성공적으로 전송된 패킷에 대한 평균 클래스 1 도착율을 나타낸다. $\lambda_{13}(pushed)$ 는 클래스 2가 높은 우선권을 가질 시간 동안, 푸시아웃된 패킷에 대한 평균 클래스 1 도착율을 나타낸다. $\lambda_{21}(eff)$ 는 클래스 1이 높은 우선권을 가질 시간 동안 성공적으로 전송된 패킷에 대한 평균 클래스 2 도착율을 나타낸다. $\lambda_{22}(pushed)$ 는 클래스 1이 높은 우선권을 가질 시간 동안, 푸시아웃된 패킷에 대한 평균 클래스 2 도착율을 나타낸다. $\lambda_{23}(eff)$ 는 클래스 2가 높은 우선권을 가질 시간 동안, 성공적으로 전송된 패킷에 대한 평균 클래스 2 도착율을 나타낸다.

그러면, 식 (11), 식 (13), 식 (14)로부터, 다음의 보조 정리를 구할 수 있다.

보조 정리 2 : IID 도착 프로세스를 가지는 유한 버퍼 시스템에 대해, 다음의 불변 관계가 성립한다.

$$\begin{aligned} &[\lambda_{11}(eff) \cdot \Pr(1 > 2) + \lambda_{12}(eff) \cdot \Pr(2 > 1) \\ &+ \lambda_{13}(pushed) \cdot \Pr(2 > 1)] ED_1(f, p) \\ &+ [\lambda_{21}(eff) \cdot \Pr(1 > 2) + \lambda_{22}(pushed) \cdot \Pr(1 > 2) \\ &+ \lambda_{23}(eff) \cdot \Pr(2 > 1)] ED_2(f, p) = C \end{aligned} \quad (15)$$

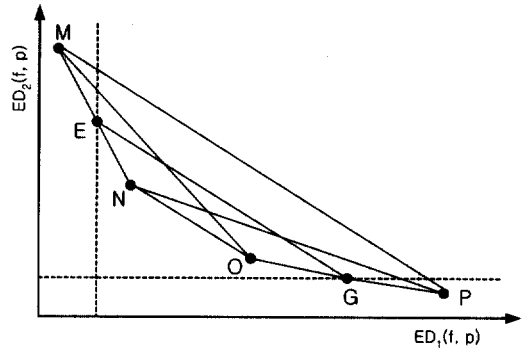
C는 정책 f와 p와는 독립적인 상수이다.

스케줄링 정책과 푸시아웃 정책을 모두 가진 패킷 지연에 대한, 식 (15)의 대표적인 보존 법칙의 형태가 (그림 17)에서 보여주고 있다. 이 그림에서, 점 E와 G는 스케줄링 정책 f만(즉, 클래스가 다른 클래스의 패킷을 푸시아웃할 수 없다) 얻을 수 있다. 식 (16)의 불변 관계는 선 EG에 대하여 성립한다.

$$\lambda_1(1 - B_1)ED_1(f) + \lambda_2(1 - B_2)ED_2(f) = C \quad (16)$$

점 E와 G는 클래스 1과 2 각각에 HOL 서비스 우선권을 줌으로써 얻어질 수 있다. 점 M은 클래스 1이 서비스 우선권을 가지고, 클래스 1보다 더 많은 푸시아웃 우선권을 클래스 2에게 줌으로써 얻어질 수 있다. 점 N은 클래스 1이

서비스 우선권을 가지고, 클래스 2보다 더 많이 클래스 1에게 푸시아웃 우선권을 줌으로써 얻어질 수 있다.



(그림 17) 스케줄링과 푸시아웃 정책을 가지는 패킷 지연의 성취 가능한 지역

$f(i > j)$ 를 클래스 j보다 i에게 높은 서비스 우선권을 주는 정책을 표시하는 것으로 정의하자. 또한 $p(i > j)$ 는 버퍼가 가득 차고 클래스 i 패킷이 도착할 때마다 클래스 j의 패킷이 항상 푸시아웃되는 정책을 표시한다고 하자. 그러면, 선 MN 상의 패킷 지연은 고정 서비스 정책 $f(1 > 2)$ 를 가질 때, $p(1 > 2)$ 와 $p(2 > 1)$ 정책간의 간단한 임의화(randomization)에 의해 얻어질 수 있다.

점 P는 서비스 우선권 $f(2 > 1)$ 를 가지고, 클래스 2보다 더 많은 푸시아웃 우선권을 클래스 1에게 줌으로써 얻어질 수 있다. 점 O는 서비스 우선권 $f(2 > 1)$ 를 가지고, 클래스 1보다 더 많이 클래스 2에게 푸시아웃 우선권을 줌으로써 얻어질 수 있다. 선 OP 상의 패킷 지연은 고정 서비스 정책 $f(2 > 1)$ 를 가질 때, $p(1 > 2)$ 와 $p(2 > 1)$ 정책간의 임의화에 의해 얻어질 수 있다.

선MO에 대한 패킷 지연은 고정 푸시아웃 우선권 $p(2 > 1)$ 를 가질 때, $f(1 > 2)$ 와 $f(2 > 1)$ 정책간의 임의화에 의해 얻어질 수 있다. 또한 선 NP에 대한 지연은 고정 푸시아웃 우선권 $p(1 > 2)$ 를 가질 때, $f(1 > 2)$ 와 $f(2 > 1)$ 정책간의 임의화에 의해 얻어질 수 있다.

(그림 17)에서 네 개의 점 M, N, O, P를 가진다. 고정 도착 통계치에 대해 버퍼 크기가 증가함에 따라, 더 작은 패킷이 손실될 것이기 때문에, 선분 MN과 선분 OP는 줄어든다. 위의 주장과 비슷하게, 선분 NO와 선분 MP상의 모든 점은 성취될 수 있다. 그러므로 어떤 정책 f와 p를 사용함으로써 사변형 MNOP내의 모든 점은 성취될 수 있다. 그러므로, 다음의 정리를 얻을 수 있다.

정리 2 : 스케줄링 정책 f와 푸시아웃 정책 p를 가진 유한 버퍼에 대하여, 패킷 지연에 대한 보존 법칙은 사변형 지역으로 표현된다.

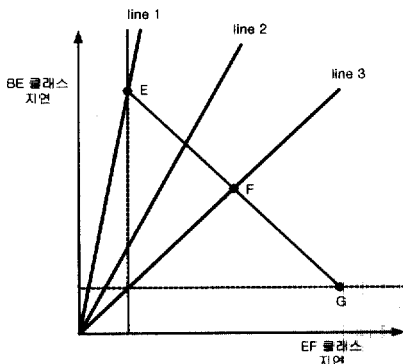
증명 : 선분 EG에 해당하는 도착율이 클래스 1과 2에 대하여 각각 λ_1 과 λ_2 라 하자. 점 M에서, $\lambda_1' < \lambda_1$ 이기 때문에, $\lambda_1(p(2 > 1)) = \lambda_1' \neq \lambda_1$. 비슷하게, $\lambda_2(p(2 > 1)) = \lambda_2' \neq \lambda_2$. 그래서,

새로운 집합의 도착을 λ_1 과 λ_2 을 가지고, 푸시아웃 기법이 없는 다른 선을 유도할 수 있다. 그것은 선분 EG와 교차점을 가진다. 비슷하게, 선분 OP로부터(점 G는 제외하고), 선분 EG와 교차점을 가지는 다른 선을 그릴 수 있다. 그러므로, 선분 MN과 OP는 선분 EG 상에 존재할 수 없다.

5.2 동적 행위 분석

본 장에서는 차등 서비스의 EF와 BE 클래스간의 할당되는 가중치에 따른 서비스 메커니즘의 동적 행위를 분석한다. 차등 서비스는 기존의 인터넷에서 채택하고 있는 최선-노력 서비스가 아닌 서비스의 종류에 따라서 QoS를 차별화하여 서비스하는 것이 기본적인 원칙이다. 이것은 앞장에서 기술한 보존 법칙으로 잘 설명될 수 있다. 어떤 보존 법칙도 “풀에게 갓기 위하여 피터에게서 빌린다”는 원칙을 적용함으로써 직관적으로 묘사될 수 있다. 즉, 한 개 이상의 트래픽 클래스를 가진 큐잉 시스템에서, 한 클래스의 트래픽에 주어지는 우선적인 취급은 성능 측도에 관계없이 다른 클래스에서의 비용으로 주어진다라는 것이다.

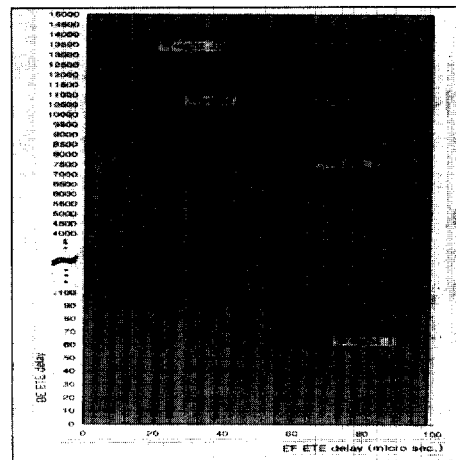
바로 이것이 차등 서비스에서 적용하는 기본적인 원칙이다. 즉, 똑같이 주어지는 서비스가 아니라 사용자에게 따라서 서비스를 차별화 하는 것이다. 적용되는 트래픽 스트림에 따라서 약간의 차이는 예상되지만, 차등 서비스 메커니즘의 시스템 행위의 기하학적 표현을 (그림 18)과 같이 기본적으로 나타낼 수 있다. 이 평면상에서, 작업 보존 정책에 대한 성취할 수 있는 평균 지연의 집합은 선분 EG로 나타낼 수 있다(여기서는 서비스 정책 f만 고려하였다). 여기서 점 E는 EF 클래스에 대한 최소 지연점을 보여주고, 점 G는 BE 클래스에 대한 최소 지연점을 나타낸다. 선 l은 지연에 대한 목표를 나타낸다. 세 개의 선 중에서, 선 1이 EF에 대하여 가장 우선 순위를 많이 준 것이고, 선 3(45도각도)은 EF와 BE 클래스에 대하여 서비스 차등을 두지 않은 최선-노력 서비스에 해당한다.



(그림 18) 시스템 행위의 기하학적 표현

(그림 19)는 가중치 1, 3, 5, 8, 10으로 할당했을 때의 EF와 BE 트래픽에 대한 종단간 지연을 모의실험을 통하여 200개의 패킷을 모아 한 배치(batch)로 하였을 때, 배치별

평균 지연을 보여준다. (본 실험에서는 각 입력 파라미터마다 30개 이상의 배치 평균을 얻을 때까지 모의실험을 수행하였다.) 이 그림에서도 EF 클래스에 가중치를 많이 줄수록 EF 클래스의 지연은 감소하는 반면(즉, (그림 18)에서의 점 E에 근접), BE 클래스의 지연은 증가되는 것을 보여준다. 가중치가 1인 경우(즉, 최선-노력 서비스에 해당), EF와 BE 클래스의 지연은 (그림 18)의 F 점에 해당하는 지역에 집중되어 있음을 확인 할 수 있다((그림 19)에서는 표현 상 수직축의 값이 크게 나타나기 때문에 45도 각도에 나타나지 않음). DiffServ 네트워크가 실제 운영될 때, (그림 18)과 같은 성능 목표에 따라 네트워크가 운영되도록 가중치의 값을 동적으로 튜닝할 수 있을 것으로 본다.



(그림 19) 가중치 인자에 의한 EF 및 BE 종단간 지연의 기하학적 표현

6. 결 론

본 논문에서는 다양한 입력 트래픽 모델 하에서 DiffServ 서비스 메커니즘의 성능평가를 위하여 인터넷 트래픽의 특성을 분석하고, 트래픽을 특성화할 수 있는 트래픽 모델링에 대한 연구를 수행하였다. 이를 기반으로, DiffServ 시뮬레이터를 설계하고 구현하여 DiffServ의 QoS 보장 기술에 대한 성능 평가를 수행하였다. 이를 위하여 먼저 트래픽 모델을 크게 자기유사 모델과 비자기유사모델로 구분하였으며, 자기유사 모델과의 비교를 목적으로 비자기유사 트래픽 모델로 랜덤 트래픽 모델과 IBP 트래픽 모델을 사용하였다. 자기유사트래픽 모델의 구현을 위하여, 중첩된 On/Off 프로세스를 이용한 자기유사 트래픽 발생기를 설계하고 시뮬레이션에 사용하였다.

개발된 DiffServ 시뮬레이터를 통하여 앞에서 기술한 랜덤, 버스티, 자기유사 입력 트래픽 모델을 사용하였을 경우에 대하여, 다양한 입력 파라미터에 대하여 모의실험을 수행하고 성능평가를 수행하였다. 성능분석 결과, 모든 환경에서 DiffServ 기능을 이용하여 EF와 BE 클래스의 서비스 분리가 가능함을 확인할 수 있었다. 그러나 BE 클래스에 대하여

는 상황에 따라서 지연이 과다하게 발생함으로 인터넷망의 입력 단에서 수락 제어 등을 통한 트래픽 제어가 필요한 것으로 분석된다. 그리고 입력 트래픽 모델에 따라서 지연과 손실 성능에 많은 차이가 남을 알 수 있었다. 따라서, 지금까지 국내에서 네트워크의 트래픽 모델로 많이 사용되어 오고 있는 포와손 트래픽을 사용하였을 때, 네트워크의 성능을 너무 낙관적으로 평가하는 결과를 초래할 것으로 분석된다. 아울러, 입력 소스에서 트래픽 특성을 평활화하기 위하여 웨이핑과 같은 기능이 필요할 것으로 분석된다.

본 논문에서는 또한 트래픽 제어의 타당성 연구를 위하여, 트래픽 스케줄링의 이론적인 근거가 되는 보존 법칙에 대하여 기술하였다. 기존에 알려진 지연과 손실 법칙은 한 개의 선분으로 표시되는 데 비하여, 스케줄링 정책과 손실 정책 모두를 가지는 DiffServ를 위한 지연 보존법칙은 4변형으로 나타남을 발견하였다. 이 개발된 보존 법칙을 기반으로 DiffServ의 자원 할당(즉, 가중치)에 대한 동적 행위를 분석하였다. 이 결과, WFQ에서의 가중치에 대하여 망의 상태를 고려한 튜닝 작업이 절대적으로 필요한 것으로 분석되었다. 또한 보다 나은 QoS 보장을 위하여 트래픽 패턴을 고려한 네트워크 설계가 중요할 것으로 판단된다.

인터넷망에서의 성능 평가는 복잡한 트래픽 특성과 망의 대규모화로 인하여 상당히 어려운 문제이다. 본 논문에서는 인터넷 트래픽의 특성을 잘 묘사할 수 있는 자기유사 트래픽 모델을 고려하였으며, 노드 레벨이 아닌 네트워크 레벨에서 시뮬레이션을 수행하였다는 것이 기존의 연구와 차이가 있다고 하겠다. 그리고 성능 분석과 합성 문제를 같이 다루었다.

향후 연구 과제로는 다양한 네트워크 트래픽 환경에서 실제 네트워크에 보다 유사하게 모의실험 환경을 구축하는 것이고, 수락 제어 등을 포함하는 통합 트래픽 제어 프레임워크를 개발하는 것이다.

참 고 문 헌

[1] X. Xiao, et al., "Internet QoS : A Big Picture," IEEE Network, pp.8-18, March/April, 1999.
 [2] R. Braden, et al., Resource ReSerVation Protocol (RSVP) -Version 1 Functional Specification. RFC 2205, Sept. 1997.
 [3] S. Blake, et al., "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.
 [4] Y. Bernet, et al., "End-to-end QoS for Differentiated Services," Internet Draft, Feb. 1999.
 [5] J. Heinanen, et al., "Differentiated Services in MPLS Networks," Internet Draft, June, 1999.
 [6] K. Nichols, et al., "Definition of the Differentiated Services Field(DS Field) in the IPv4 and IPv6 Headers," RFC 2474, Dec. 1998.
 [7] J. Heinanen, et al., "An Assured Forwarding PHB Group," RFC 2597, June, 1999.
 [8] V. Jacobson, et al., "An Expedited Forwarding PHB," RFC 2598, June, 1999.
 [9] Loukola M. V., et al., "Dynamic RT/NRT PHB group,"

Internet Draft<draft-loukola-dynamic-00.txt>, Nov. 1998.
 [10] William Stallings, High-Speed Networks : TCP/IP and ATM Design Principles, Prentice-Hall, 1998.
 [11] I. Norros, "A storage model with self-similar input," Queueing System, 16 : pp.387-396, 1994.
 [12] P., Ramakrishnan, "Self-similar traffic Models," CSHCN Tech. Report, pp.99-5, 1999.
 [13] A. Erramilli et al., "Chaotic maps as models of packet traffic," ITC-14, 1994.
 [14] http://ita.ee.lbl.gov/.
 [15] 박정숙, 자기유사 트래픽 특성을 고려한 차세대 인터넷 망의 QoS에 대한 성능 평가, 대구가톨릭대학교 대학원 전산통계학과 박사학위 논문, 2001.
 [16] 전용희, "ATM 통신망에서의 지연 및 손실 우선순위를 갖는 다중화 알고리즘의 성능평가", 한국통신학회논문지 제19권 제5호, pp.842-856, 1994.
 [17] Y. Bernet, et al., "A Conceptual Model for DiffServ Routers," Internet Draft, June, 1999.
 [18] 전용희, 박수영, "DiffServ를 이용한 인터넷 QoS 보장 기술", 한국통신학회지, 제17권 제9호, pp.1152-1173, 2000.
 [19] E. Gelenbe and I. Mitrani, Analysis and Synthesis of Computer Systems, New York, Academic Press, 1980.



전 용 희

e-mail : yhjeon@cataegu.ac.kr
 1978년 고려대학교 전기공학과(공학사)
 1987년 미국 플로리다공대 대학원 컴퓨터공학과(수료)
 1989년 미국 노스캐롤라이나 주립대 Elec. and Comp. Eng. 석사(MS)
 1992년 미국 노스캐롤라이나주립대 Elec. and Comp. Eng. 박사(Ph.D)

1978년~1978년 삼성중공업(주) 근무
 1978년~1985년 한국전력기술(주) 근무
 1989년~1989년 미국 노스캐롤라이나주립대 Elec. and Comp. Eng. TA
 1989년~1992년 미국 노스캐롤라이나주립대 부설 CCSP(Center for Comm. & Signal Processing) RA.
 1992년~1994년 한국전자통신연구소 통신시스템연구단 선임 연구원
 1994년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 학부장(현), 공과대학장(현)
 관심분야 : 차세대 인터넷, 통신망 성능분석, 초고속망 프로토콜, 통신망 보안



박 수 영

e-mail : sypark@mail.uiduk.ac.kr
 1991년 대구가톨릭대학교 전산통계학과 학사
 1996년 대구가톨릭대학교 대학원 전산통계학과 석사
 1999년 대구가톨릭대학교 대학원 전산통계학과 박사과정 수료

2002년 대구가톨릭대학교 대학원 전산통계학과 박사
 1990년~1992년 대구백화점(주)정보시스템부 근무
 1996년~1999년 대구미래대학 멀티미디어정보과학과 겸임교수
 2001년~현재 위덕대학교 컴퓨터공학과 전임강사
 관심분야 : 차세대 인터넷, QoS 보장기술, 고속 통신망 응용 서비스, 통신망 보안