

트래픽 모델과 Weight Factor에 의한 차등 서비스 메커니즘의 성능평가

종신회원 전 용 희*, 준회원 박 수 영**

Performance Evaluation of a Differentiated Service Mechanism by Traffic Models and Weight Factor

Yong-Hee Jeon*, Soo-Young Park** *Regular Members*

요 약

현재의 인터넷망에서 채택하고 있는 최선노력 서비스만으로는 실시간 응용 서비스들이 요구하는 서비스 품질(QoS: Quality of Service) 보장을 만족시키는 것이 어렵다. 따라서 세계적으로 QoS를 제공하기 위한 방법에 대한 연구가 진행되고 있다. 그 중에서 본 논문에서는 차등 서비스(DiffServ: Differentiated Service)를 이용한 QoS 보장 기법에 대하여 기술하였다. 먼저 차등 서비스의 성능이 어떻게 트래픽 모델에 의하여 영향을 받는지 분석하였다. 이를 위하여, 랜덤, 버스티, 자기유사 입력 트래픽 모델링 방법에 대하여 연구를 수행하였다. 그리고 OPNET 시뮬레이터를 설계 및 구현하고 다양한 입력 파라미터에 대하여 모의실험과 성능평가를 수행하였다. 성능분석 결과 모든 환경에서 DiffServ 기능을 이용하여 EF(Expedited Forwarding) 클래스에 대하여 QoS 보장이 가능하고, EF와 BE(Best Effort) 클래스의 서비스 분리가 가능함을 확인할 수 있었다. WFQ(Weighted Fair Queueing)에서 EF와 BE 클래스 사이에 주어진 자원의 분배(즉, Weight factor)에 따른 성능 변화와 역동적 행위에 대하여도 분석하였다.

ABSTRACT

It is difficult to satisfy the QoS(Quality of Service) guarantee which is required by real-time application services only with the Best-Effort(BE) service adopted in the current Internet. Therefore, worldwide research is being made on the method of QoS provisioning. Among them, the QoS guarantee mechanism using the DiffServ(Differentiated Service) was discussed in this paper. First we analyzed how the DiffServ performance was affected by traffic models. For this, we performed the research for the random, bursty, and self-similar traffic modeling method. We then designed and implemented an OPNET simulator, and performed the simulation and performance evaluation for diverse input parameters. Based on the results of performance evaluation, it was confirmed that QoS guarantee is possible for the EF(Expedited Forwarding) class with the DiffServ function under every environments considered and the service separation between EF and BE(Best Effort) classes is also possible. We also analyzed the performance variation and dynamic behavior of DiffServ mechanism based on the resource allocation between EF and BE classes in WFQ(Weighted Fair Queueing).

I. 서 론

최선-노력(Best-Effort) 서비스만으로 패킷을 전달

하는 현재의 인터넷으로는 실시간 응용 서비스들이 요구하는 QoS(Quality of Service) 보장을 만족시키는 것이 불가능하다. 따라서, 다양한 트래픽에 대해

* 대구가톨릭대학교 컴퓨터정보통신공학부 (yhjeon@curh.cataegu.ac.kr)

** 위덕대학교 컴퓨터공학과 (sypark@mail.uiduk.ac.kr)

논문번호 : 010181-0713, 접수일자 : 2001년 7월 13일

서 서로 다른 QoS를 제공하기 위한 연구가 진행되고 있다. 90년대 초에 인터넷에서 QoS 보장을 제공하기 위해서 통합서비스(IntServ) 모델이 제안되었다^{11,2)}. 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 있다³⁻⁴⁾.

DiffServ 네트워크의 목표는 인터넷에서 확장 가능한 차등 서비스를 제공하는 것인데, 이는 네트워크 경계(boundary) 노드에서만 수행되는 트래픽 분류와 조절로 가능하다. 즉, 패킷 집합 분류 및 처리는 트래픽 처리가 많지 않은 에지 라우터에서 수행을 하고 코어 라우터는 패킷별 처리만으로 빠른 전달만을 수행하도록 한다. 코어 라우터에서의 패킷 처리가 단순해지게 된 것은 홉간의 패킷처리 규칙을 규정한 PHB(Per-Hop Behavior) 개념의 도입으로 가능해졌다. DiffServ에서 PHB는 한 라우터에서 다른 라우터로 트래픽을 전달하는 기본 방침이 된다.

DiffServ 네트워크의 입구 에지 라우터에서는 먼저 패킷을 분류한 다음, 분류된 패킷은 TCA(Traffic Conditioning Agreement)를 기준으로 트래픽 조절기에 의해 미터링, 마킹, 셰이핑 기능으로 처리된 IP 패킷은 DSCP(DS Code Point) 값을 헤더에 싣고 DiffServ 네트워크로 진입한다. 도착한 패킷의 DSCP에 의해 라우터는 PHB를 적용한다.

DSCP 값을 이용해 Internet Draft에 권고된 PHB 그룹으로는 Default⁶⁾, Assured Forwarding⁷⁾, Expedited Forwarding⁸⁾, Class Selector Compliant⁹⁾, Dynamic RT/NRT PHB 그룹¹⁰⁾ 등으로 분류된다. 현재 DiffServ 모델에서는 세 가지 서비스 클래스가 제안되었다: DE(Default)⁶⁾, AF(Assured Forwarding)⁷⁾, EF(Expedited Forwarding)⁸⁾ PHB. PHB의 구현을 위해서 버퍼관리와 패킷 스케줄링 메커니즘 등이 이용된다.

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.

먼저 II장에서는 인터넷 트래픽 특성과 트래픽 모델링에 대한 내용을 기술한다. III장에서는 구현된 차등 서비스 메커니즘에 대해서 기술하고, IV장에서는 구현된 차등 서비스 성능분석 시뮬레이터와 성능분석 결과를 제시한다. V장에서는 차등 서비스 메커니즘의 자원 할당에 대한 역동적 행위(Dynamic Behavior)에 대하여 분석한다. 마지막 VI장에서는 결론과 향후 연구 계획으로 글을 맺는다.

II. 트래픽 모델링

현재 제공되는 많은 인터넷 서비스 트래픽들의 특성은 기존 트래픽 성질과는 완전히 다른 자기유사성이라는 성질들을 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 증명되었다⁵⁾. 그러므로, 자기유사적인 트래픽 성질이 인터넷 네트워크의 성능에 미치는 영향을 예측 분석하기 위해 자기유사 트래픽을 정확하고 효율적으로 모델링하기 위한 방안에 대한 연구 및 분석 결과는 새롭게 설계 구현되어지는 DiffServ 네트워크에도 반드시 고려되어야 할 사항들이라 판단된다.

1. 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 비자기유사 트래픽 모델로 랜덤 트래픽 및 IBP 트래픽 모델을 사용하였다.

1.1 랜덤 트래픽

포와손 프로세스(Poisson Process)는 여러 가지 해석상의 특성 때문에 기존 전화망이나 데이터망의 트래픽 모델에 주로 사용되어 왔다. 그 이유는 다음과 같이 몇 가지가 있다.

- ① 독립적인 포와손 프로세스의 중첩은 그것의 율이 각 성분들의 합이 되는 또 하나의 새로운 포와손 프로세스가 된다.
- ② 포와손 프로세스의 메모리스(memoryless) 특성

이 포와손 프로세스를 이용하는 대기 문제(queueing problem)를 크게 단순화시킨다.

③포와손 프로세스는 많은 수의 독립적인 트래픽 스트림을 구성하고 있는 트래픽 응용에 통상적으로 적용 가능하다는 것이다.

본 논문에서는 랜덤 트래픽 모델을 포와손 프로세스의 이산버전인 베르누이 프로세스에 의해 표현한다. 베르누이 프로세스에서 각 슬롯에서의 도착 확률은 각 슬롯 사이가 독립적으로 p 이다. 그림 1은 랜덤 트래픽 모델에서의 시간 슬롯(time slot)을 보여주고 있다.



그림 1. 랜덤 트래픽 모델에서의 시간 슬롯

1.2 버스티 트래픽

지수분포를 가지는 On(active period) 상태와 또 하나의 다른 독립적인 지수분포를 가지는 Off(silent period) 상태가 고대로 나타나는 프로세스인 IPP 모델은 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 2 참조).

프로세스가 활동 상태에 있을 때 다음 슬롯에서 확률 p 를 가지고 그 상태에 머물러 있거나 확률 $1-p$ 를 가지고 휴지 상태로 이동한다. 만약 프로세스가 휴지 상태에 있다면 확률 q 를 가지고 휴지 상태에 계속 머물고 확률 $1-q$ 만큼 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^[11].

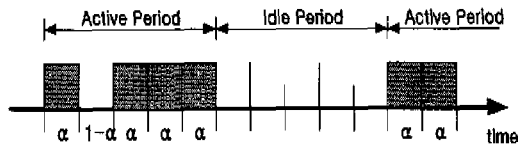


그림 2. 버스티 트래픽 모델에서의 시간 슬롯

IBP 프로세스의 전이 확률(transition probability) 행렬은 다음 식 (1)과 같다.

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

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.

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

$$\Pi_I = \frac{1-p}{2-p-q} \tag{3}$$

여기서, Π_A 는 평균 도착율을 의미한다.

d 를 연속적인 패킷들 사이의 도착간 시간(inter-arrival time)으로 정의하면, 도착간 시간의 제곱 변화 계수(squared coefficient of variation), C^2 을 다음 식(4)에 의하여 구할 수 있다.

$$C^2 = \frac{Var(d)}{[E(d)]^2} = \frac{(p+q)(1-p)}{(2-(p+q))^2} \tag{4}$$

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

2. 자기유사 트래픽 모델

자기유사성을 표현하는 트래픽 모델은 대표적으로 FGN(Fractional Gaussian Noise)^[12], F-ARIMA (Fractional Autoregressive Integrated Moving Average)^[13], Chaotic map^[14] 등이 있다.

2.1 자기유사 트래픽

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 상관이 식 (5)~(7)과 같이 정의된다.

평균: $E[x(t)] = \frac{E[x(at)]}{a^H}$ (5)

분산: $Var[x(t)] = \frac{Var[x(at)]}{a^{2H}}$ (6)

자기상관: $R_x(t, s) = \frac{R_x(at, as)}{a^{2H}}$ (7)

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

· 장기간 의존성(LRD: long-range dependency): 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 특징을 가지고 있다. 이런 특성이 네트워크의 성능분석에 심대한 영향을 미칠 수 있다.

· Heavy-tailed 분포: 패킷 도착간 시간이나 버스트 길이와 같은 트래픽 프로세스를 나타내는 확률 밀도 함수를 특성화하기 위하여 사용되며, 확률 변수 X 의 분포가 다음 식(8)을 만족하면 긴 꼬리라고 한다.

$$1 - F(x) = \Pr[X > x] \sim \frac{1}{x^a} \quad (x \rightarrow \infty, a > 0) \quad (8)$$

일반적으로, 긴 꼬리 분포를 가진 확률변수는 크거나 무한인 분산 값을 가진다.

· 스펙트럼 밀도: 주파수 영역에서 기술하면, LRD 프로세스의 전력 스펙트럼 밀도는 식 (9)와 같은 원래의 신호에 가까운 전력 법칙을 따른다.

$$S(w) \sim \frac{1}{|w|^r}, \quad (w \rightarrow 0, 0 < r < 1) \quad (9)$$

자기유사적인 트래픽을 생성하는 방법으로는 긴 꼬리 On/Off 구간들을 가지는 단순 On/Off 프로세스들을 다수 중첩시켜서 자기유사적 프로세스를 구축하는 방법, M/G/∞ 큐잉모델, RMD(Random Midpoint Displacement), Wavelet 계수 계산법, Hosking 방법, FFT(Fast Fourier Transform) 방법 등이 있다^[5]. 이처럼 자기유사 트래픽을 모델링하는 기법에는 여러 가지가 있는데, 본 논문에서는 On/Off 프로세스 기법을 이용하여 구현하였다.

1) 중첩된 On/Off 프로세스

웹을 포함하여 현재 인터넷에서 제공되어지는 서비스를 위한 파일 구조는 heavy-tailed 분포를 가지며, 이러한 특성은 합성된 네트워크 트래픽의 자기

유사성을 일으키는 원인을 제공한다. 따라서, 간단한 On/Off 프로세스를 중첩시키면 자기유사적인 결과가 나타난다. 이 때 중첩되는 스트림의 수가 증가될수록 정확한 자기유사 모델의 표현이 가능하는데, 최소 150 개 이상의 긴 꼬리 분포를 가지는 스트림의 중첩이 필요한 것으로 알려져 있다. 이러한 모델은 개별적 On/Off 소스는 데이터 통신망 터미널 모델링에 상당히 유용하다. 이 모델에서의 트래픽 소스는 On 구간동안 R 의 율로 데이터를 전송하거나 Off 구간동안 idle로 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) FGN(Fractional Gaussian Noise)

연속시간 확률 프로세스 $Z = \{Z(t), t \geq 0\}$ 는 다음과 같은 조건을 만족하면 Hurst 파라미터 H 를 가진 FBM(Fractional Brownian Motion)이다: 1) Z 는 가우스 분포이다, 2) $Z(0) = 0$, 3) Z 는 연속 표본 경로를 가진다, 4) Z 는 정상적 증분치들을 가진다, 5) $E[Z(t)] = 0, t \geq 0$, 6) $E[Z(t)^2] = \sigma^2 t^{2H}, t \geq 0$. FBM은 H 를 가지고 자기유사적이다. FBM Z 가 주어질 때 이산시간 증분 프로세스 $X = \{X_n\}_{n=0}^{\infty}$ 로 정의가 가능하며, 이때 $X_n = Z(n+1) - Z(n)$ 이다. 이런 X 를 H 를 가진 FGN이라 한다. 자기 공분산 함수 $r(l)$ 이 $0.5 < H < 1$ 에 대하여 0에 hyperbolic하게 접근하여 $\sum_{l=0}^{\infty} r(l) = \infty$ 이 되므로, LRD 성질을 지니게 된다.

3) F-ARIMA

F-ARIMA 모델의 장점은 단기 및 장기간 의존성을 동시에 포착 가능하다는 것이다. 단편적(fractional) ARIMA(p,d,q)에 대해 프로세스 $X = \{X_t; t = 1, 2, \dots\}$ 는 식 $\phi(B) \nabla^d X_t = \theta(B) \varepsilon_t$ 로 정의되며, 파라미터 d 는 0과 1/2 사이에 있고, $\nabla^d = (1-B)^d$ 는 식 $(1-B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-1)^k B^k$ 으로 표현 가능하다. 파라미터 d 는 장기간 의존성을 나타낸다. 그리고 p 와 q 는 단기간 행위를 나타내며, $H = d + 1/2$ 이다. F-ARIMA 모델은 연속적 관측치 사이에 변화가 많지 않는 시퀀스의 모델링에 유용하다.

III. 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. 즉 DiffServ 모델에서는 DiffServ 라우터가 핵심 노드라고 할 수 있다. DiffServ 라우터에는 에지 라우터와 코어 라우터가 있다^[15]. 본 논문에서는 에지 라우터에 중점을 둔다.

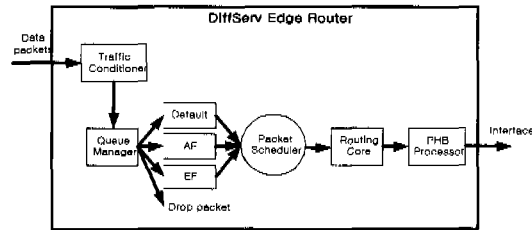


그림 3. DiffServ 라우터의 구성요소

그림 3에서 보여주는 DiffServ 라우터의 요소는 트래픽 분류(Classifier), 미터링(Metering), 마킹(Marking), 셰이핑(Shaping) 기능으로 이루어진 트래픽 조절(Traffic Conditioning: TC)과 큐 관리 및 트래픽 스케줄링 기능으로 구성되어 있다. DiffServ 라우터 구성요소의 각 기능은 다음과 같다.

1) 분류

분류는 DiffServ 라우터로부터 요구되는 가장 기본적인 기능이다. 패킷 분류기는 IP 헤더에 있는 DSCP(DiffServ CodePoint)만을 기초로 패킷을 분류하는 BA(Behavior Aggregate) 분류기와 근원지/목적지 주소, 프로토콜 ID, 근원지/목적지 포트번호, 입력 인터페이스 등과 같은 한 개 이상의 헤더필드 조합 값을 기초로 패킷을 선택하게 되는 MF(Multi-Field) 분류기가 있다. 분류된 후 패킷은 해당 클래스에 입력된다.

2) 미터링

미터링은 분류기에 의해 선택되어진 트래픽 스트림상의 패킷 도착율을 확인하여 TCA(Traffic Conditioning Agreement)의 트래픽 프로파일에 일치하는지를 결정해 in-profile/out-of-profile을 수행하는 기능이다.

3) 마킹/리마킹

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.

4) 드롭퍼/셰이퍼

만약 EF 패킷이 out-of-profile이라면 드롭퍼에 의

해 즉시 탈락된다. 셰이핑은 지터를 제거하기 위해 에지 노드나 경계 노드에서 발생한다.

5) 큐잉 정책

마지막으로 큐잉 요소는 PHB에서 사용되는 큐잉 스케줄링을 적용시키는 부분이다. 실제로 차별화가 이루어지는 부분이다. 다음과 같은 큐잉 스케줄링 알고리즘이 사용된다.

- ①FIFO : 가장 단순한 알고리즘으로 출력 큐에 맨 처음 있는 패킷이 가장 먼저 전송되는 방식이다.
- ②WFQ(Weighted Fair Queuing) : 각 트래픽 흐름에 가중치를 할당하여 자동 서비스를 제공할 수 있는 알고리즘으로 독립적인 큐를 가지며 특수시에도 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 요소로는 TCB, 버퍼 관리, 스케줄러의 요소로 이루어져 있다. TCB는 트래픽 분류기능과 조절기능이 조합되어 있는 블록이다. 트래픽 분류기는 BA 분류기를 사용하여, DSCP 필드를 이용하여 패킷을 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.

IV. 시뮬레이션 모델 및 성능분석

1. DiffServ 시뮬레이터

1.1 시뮬레이터 설계 및 구현

이 절에서는 소프트웨어 도구 OPNET을 사용해, DiffServ 성능을 모의실험 할 시뮬레이터를 구현하였다. 패킷 크기는 512 바이트(Byte)이며, 패킷 전송율은 155(Mbps)로 설정했다. DiffServ 네트워크가 올바르게 동작하는지 검사하기 위해서 평균 종단간 지연 통계치(Average End-to-End delay statistic)를 이용하였는데, 이 통계치는 시스템의 성능을 분석하고 개선을 연구하기 위해 사용되었다.

먼저 DiffServ 시뮬레이터에서 구현된 트래픽 조절 기능들을 설명한다.

1) 트래픽 조절 기능 구현

(1) 분류(Classifying)

BA 분류는 코드포인트만으로 분류하여 구현되었

다. 패킷간 차별화를 만들기 위해 IPv4 헤더에 있는 DSCP에 접근하도록 만들었다. 그림 4는 DSCP 필드(DS byte)를 포함하는 IP 패킷 필드를 보여준다.

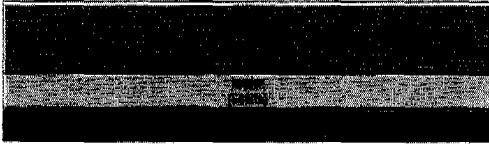


그림 4. DS CodePoint를 가지는 IP 패킷 필드

(2) 마킹(Marking)

도착하는 패킷은 코드포인트의 클래스 선택자에 의해 트래픽 클래스 EF, BE 트래픽으로 분리된 후, 마킹 되고 큐잉된다. 그림 5는 TCB의 프로세스 모델의 알고리즘을 보였고, “classifier” 상태에서 입력 스트림을 통해 입력되는 패킷을 BA 분류 방법으로 분류하고, “marking” 상태에서 marker에 의해 DSCP에 마킹을 한다.

```

/* classify & marking */
op_pk_nfd_get(pkptr, "tos", &tos_value);
switch(tos_value)
case tos : Classify to PHB Class;
           DSCP marking;
           break;
    
```

그림 5. 분류 및 마킹 프로세스 모델 알고리즘

(3) 큐잉(Queueing)

DiffServ 구조의 장점을 얻는데 큐잉 알고리즘은 매우 중요하다. 여기서는 FIFO와 WFQ 스케줄링 알고리즘이 시스템에서 구현되었다. 그림 6은 WFQ 큐잉 프로세스 모델을 위한 간단한 알고리즘을 보여준다. 마킹된 패킷들은 해당하는 각 큐에 입력이 된다. 그 다음 WFQ 트래픽 스케줄링을 사용해, 가중치를 부여받은 패킷 스트림은 차등화된 대역을 할당받는다.

```

pkptr=op_pk_get(op_intrpt_strm());
switch(DSCP)
{
case BE: Insert into low weight Queue;
case EF: Insert into high weight Queue;
}
Service according to each Queue weight
    
```

그림 6. WFQ 프로세스 모델 알고리즘

2) 패킷 생성기 구현

(1) 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. 그림 7은 랜덤 트래픽 생성기 프로세스 모델 알고리즘을 보여준다.

```

Generate a random number r;
if(r <= traffic_load)
{
pkptr=op_pkt_create_fmt("ip_header");
/* Generate Packet with ToS field */
op_pk_nfd_set(pkptr, "tos", "tos_value");
op_pk_send(pkptr, next_node);
}
    
```

그림 7. 랜덤 트래픽 생성기 프로세스 모델 알고리즘

(2) 버스티 트래픽 모델

버스티 트래픽의 생성은 IBP에 의하여 수행된다. IBP 프로세스에서의 사용자 입력 파라미터는 트래픽 부하량과 버스티니스(burstiness)의 값, 그리고 활동기간 중에 있는 각 C^2 시간 슬롯마다 패킷을 포함할 확률인 α 이다. 이런 파라미터의 값을 다양하게 설정해 버스티 특성을 가진 트래픽에 대해서는 DiffServ 시뮬레이터가 어떤 성능을 나타내는지 분석하였다. 그림 8은 버스티 트래픽 모델에 의해 구현된 패킷 생성기 프로세스에서 작동하는 버스티 트래픽을 발생시키는 알고리즘을 나타내었다.

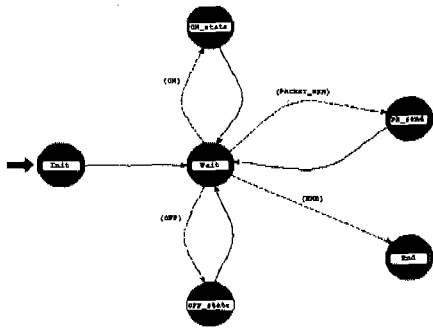
```

Calculate rate;
Generate a random number r;
if(r < rate)
Stay in the same state;
else
Move to the other state;
if (active state) {
Generate a random number gen_rate;
if (gen_rate <= alpha) {
pkptr=op_pkt_create_fmt("ip_header");
op_pk_nfd_set(pkptr, "tos", "tos_value");
op_pk_send(pkptr, next_node);
}
}
    
```

그림 8. 버스티 트래픽 생성기 알고리즘

(3) 자기유사 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 9는 자기유사 트래픽 모델에 의해 구현된 패킷 생성기 프로세스를 알고리즘과 함께 나타내었다.



```

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 (gem == 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;
}

pkptr=op_pkt_create_fmt("ip_header");
op_pk_nfd_set(pkptr, "tos", "tos_value");
op_pk_send(pkptr, next_node);
    
```

그림 9. 자기유사 트래픽 생성기 프로세스 모델과 알고리즘

2. 성능분석 시나리오

DiffServ 기능 검증을 위하여 앞 절에서 구현된 DiffServ 시뮬레이터를 사용하여 DiffServ 네트워크를 구현해 인터넷 QoS 지원 기능에 대한 성능 평가를 수행할 필요가 있다. 그래서 성능분석을 위한 시나리오를 세 가지 만들어 DiffServ 시뮬레이터를 이용하여 모의실험을 수행하고 성능 평가를 수행하였다.

2.1 시나리오에 적용되는 트래픽 스트림

여러 가지 시뮬레이션 시나리오를 통해 DiffServ 기법의 성능을 분석하려고 하였다. 각 시나리오를 표 1로 요약했다.

표 1. 성능분석을 위한 시나리오

환경 시나리오	트래픽 모델	PHB	큐잉 알고리즘	라우터 개수	통신 모드
시나리오 1	랜덤 버스티 자기유사	EF BE	FIFO	3	단방향
시나리오 2	랜덤 버스티 자기유사	EF BE	WFQ	3	단방향
시나리오 3	랜덤 버스티 자기유사	EF BE	WFQ	5	단방향

시나리오 1은 다른 DiffServ 시나리오의 장점을 보여주기 위해 만들어졌다. 이 모델에서는 TCB가 없고, 오로지 한 개의 큐만이 모든 입력 트래픽을 받아들인다. 적용되는 트래픽 스케줄링 정책은 FIFO이다. DiffServ 시뮬레이터가 적용될 시나리오 2에서는 앞 절에서 설명된 DiffServ 기능을 가진 라우터가 사용된다. 라우터는 세 개가 사용되며, 코어 라우터는 트래픽을 분류, 마킹, 재분류하지 않고 포워딩 기능만을 가지므로 보다 작은 큐와 작은 처리 지연을 가지지만, 에지 라우터에서는 DiffServ 기능을 가진 TCB 기능이 사용된다.

그림 10은 동일한 DiffServ 시뮬레이터가 적용되지만 라우터 개수를 시나리오 2에서보다 2개 더 확장을 시킨 시나리오 3을 보여준다. 다른 환경 설정은 시나리오 2와 동일하다.

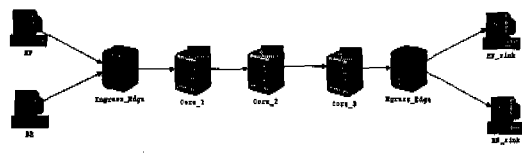


그림 10. 시나리오 3

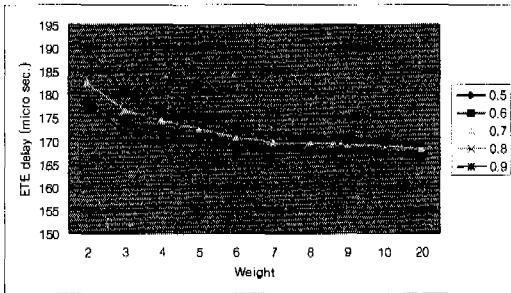
3. 성능분석 결과

본 장에서는 차등 서비스를 지원할 수 있는 시뮬레이터를 개발하여 성능 모의실험 결과를 제시하고 분석하였다. 구현된 시뮬레이터는 DiffServ 네트워크의 핵심인 DiffServ 라우터의 성능 분석을 위하여 사용될 수 있다. 이 시뮬레이터에서는 버스티 트래픽 뿐 아니라 현재의 인터넷 트래픽 특성으로 나타난 자기유사 트래픽에 대해서도 차등 서비스의 QoS 지원 기능을 성능 평가할 수 있다. 그러므로 본 논문에서는 구현된 각 트래픽 모델(즉, 랜덤, 버스티, 자기 유사 모델)에 대해 EF 및 BE PHB 두 가지 트래픽 클래스로 차별 서비스에 대한 모의실험을 수행하였다. 본 논문에서는 지면관계 상 시나리오 3에 대하여만 기술한다.

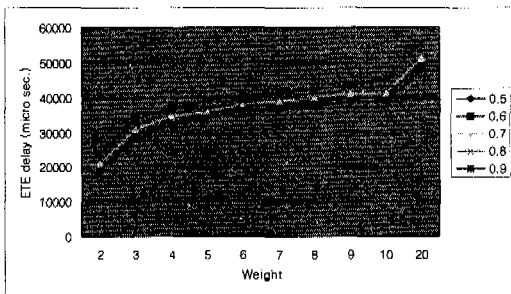
3.1 랜덤 트래픽에 대한 결과 및 분석

1) 평균 종단간 지연

그림 11의 (a)와 (b)는 트래픽 부하량 50%에서 90%까지에 대하여 EF와 BE 트래픽의 평균 종단간 지연을 보여주고 있다. 여기서는 Weight가 높을수록 EF 트래픽의 평균 종단간 지연이 감소하고 있는 것을 보여주고 있다. 반면 BE 트래픽의 평균 종단간 지연은 증가하는 것을 알 수 있다. 즉, EF와 BE 서비스의 분리로 현재 인터넷에서는 제공되지 않는 차등화 서비스가 가능함을 보여준다.



(a) EF Traffic



(b) BE Traffic

그림 11. weight 값에 따른 평균 종단간 지연

EF 트래픽은 모든 부하에 대해서 Weight 8 이후는 거의 조금씩 감소하고 있는 것을 보여주며, BE 트래픽은 Weight 10까지는 서서히 증가하지만, 그 이후는 급격히 증가하고 있다. Weight 10 이상의 값에서는 EF 트래픽의 평균 종단간 지연이 줄어드는 것에 비해 BE 트래픽의 평균 종단간 지연이 증가하는 것을 볼 수 있다. 즉, EF 트래픽이 얻는 지연의 감소에 비하여 BE 트래픽의 종단간 지연이 더 많이 증가됨을 알 수 있다. 그러므로, WFQ 큐잉 스케줄링에서 너무 큰 값의 Weight 할당은 좋지 않는 것으로 분석할 수 있다. 네트워크 부하에 따른 Weight 할당에 관한 튜닝 작업도 DiffServ 네트워크에서는 아주 중요한 작업으로 판단된다. 그림 11에서 BE 클래스의 지연은 전체적인 트래픽 인가부하 보다 Weight의 값에 더 많은 영향을 받는 것으로 또한 분석된다.

그림 12는 EF와 BE 트래픽의 평균 종단간 지연을 한 그래프 내에 나타내었다. 이 그림에서는 EF와 BE 서비스간의 지연이 확실히 차별화되는 것을 비교할 수 있다. 높은 전체 인가 부하에서도 EF 서비스는 QoS가 보장되는 반면에 BE 트래픽의 종단간 지연은 상대적으로 크게 증가됨을 볼 수 있다.

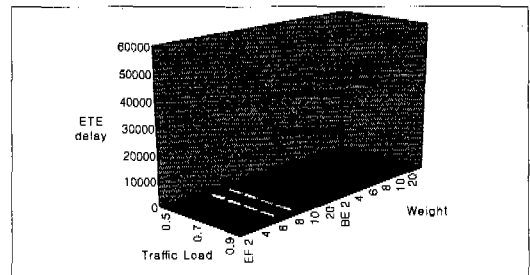
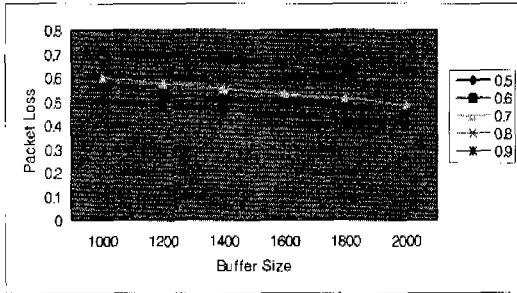


그림 12. EF와 BE 서비스의 평균 종단간 지연 비교

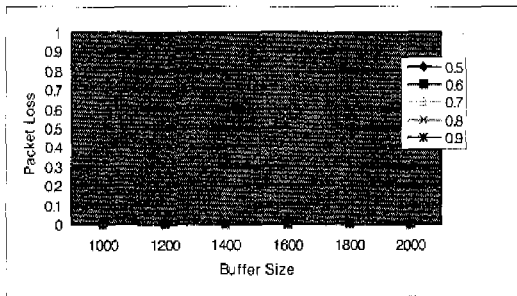
2) 패킷 손실

그림 13의 (a)와 (b)는 트래픽 부하량 50%에서 90%까지에 대하여 EF와 BE 트래픽의 패킷 손실을 보여주고 있다. Weight는 5로 설정을 했고, 버퍼 크기에 따라 패킷이 얼마만큼 손실이 발생하는가를 조사하였다. 버퍼 크기는 1000에서 2000까지의 범위로 설정을 했다. BE 트래픽의 경우는 많은 패킷 손실이 발생했는데, 예상되는 바와 같이 버퍼 크기가 증가할수록 패킷 손실률이 감소되는 것을 볼 수 있다. 그러나 EF 트래픽의 경우는 버퍼 크기에 상관없이 전혀 패킷의 손실이 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 된다. 그러므로 손실이 많이 발생한다. 따라서, 버퍼 크기를 결정할 때 이런 점을 고려하여, BE 클래스에 대하여는 EF 보다 훨씬 더 큰 버퍼를 할당해야 할 것으로 분석된다.



(a) BE Traffic



(b) EF Traffic

그림 13. 버퍼 크기에 따른 패킷 손실

그림 14는 EF와 BE 트래픽의 패킷 손실 정도를 한 그래프 내에 나타내었다. 이 그림에서는 EF와 BE 서비스간의 패킷 손실이 확실히 차별화되는 것을 비교할 수 있다. 높은 전체 인가 부하에서도 EF 서비스는 QoS가 보장되는 반면에 BE 트래픽의 패

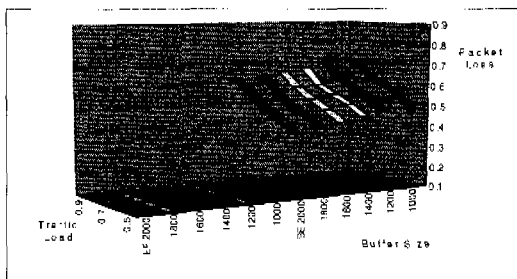


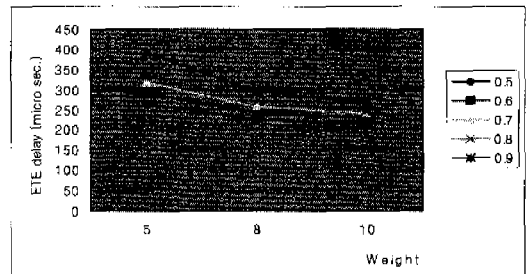
그림 14. 랜덤 트래픽에 대한 EF와 BE 서비스의 패킷 손실 비교

킷 손실은 패킷 크기가 증가함에 따라 패킷 손실이 감소되기는 하지만 EF에 비해 상대적으로 많이 발생함을 볼 수 있다.

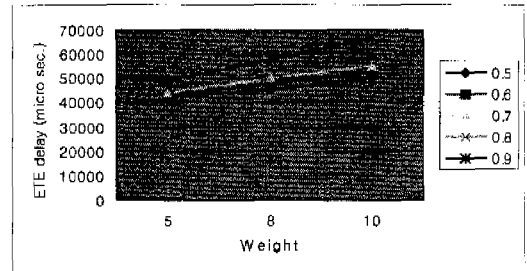
3.2 버스티 트래픽에 대한 결과 및 분석

1) 평균 종단간 지연

그림 15 (a)와 (b)는 버스티니스의 척도인 C^2 값을 20으로 했을 때, 트래픽 부하량 50%에서 90%까지에 대하여 EF와 BE 트래픽의 평균 종단간 지연을 보여주고 있다.



(a) EF Traffic



(b) BE Traffic

그림 15. 버스티 입력 트래픽에 대한 종단간 지연

랜덤 트래픽 모델을 이용한 DiffServ 시뮬레이터 모의 실험 결과 10이 넘는 Weight 값의 할당은 BE 서비스에 과다한 지연을 초래하는 것으로 결과가 나왔기 때문에 버스티 트래픽 모델에 대해서는 Weight 5, 8, 10에 대한 결과만 그래프로 나타내었다.

버스티 트래픽에 대해서도 랜덤 트래픽과 마찬가지로 Weight가 높을수록 EF 트래픽의 평균 종단간 지연이 감소하는 반면 BE 트래픽의 평균 종단간 지연은 증가하는 것을 볼 수 있다. 그러나 랜덤 트래픽의 종단간 지연보다는 지연 값이 많이 높아진 것을 알 수 있다. 이것은 대역폭 보장을 받지 못하는 BE 서비스의 지연 특성이 트래픽의 버스트 성으로 악화되기 때문이다.

그림 16은 버스티 트래픽에서 EF와 BE 클래스의 평균 종단간 지연을 동일한 그래프 영역에서 비교한 것이다. 버스티 입력 트래픽인 경우에도 EF 클래스에 대하여는 종단간 QoS가 보장됨을 보여준다. 그러나, 버스티니스의 값이 커질수록 BE 클래스의 지연은 크게 증가된다.

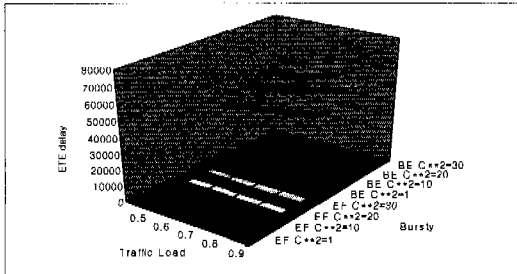
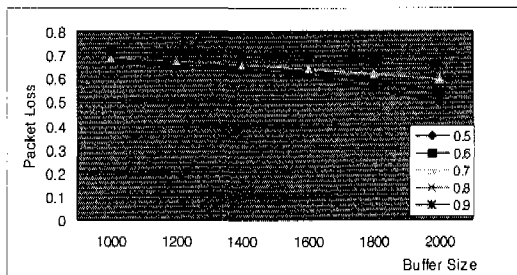


그림 16. 버스티 입력 트래픽에 대한 EF와 BE 클래스의 평균 종단간 지연 비교

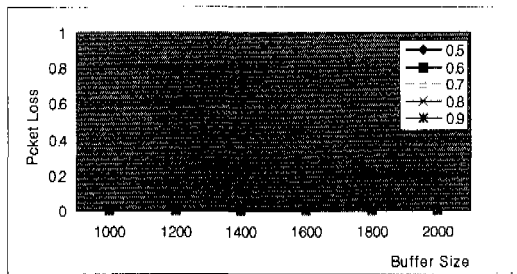
2) 패킷 손실

그림 17 (a)와 (b)는 버스티니스의 척도인 C^2 값을 20으로 했을 때, 트래픽 부하량 50%에서 90%까지에 대하여 EF와 BE 트래픽의 패킷 손실을 보여주고 있다.

버스티 트래픽에 대해서도 랜덤 트래픽과 마찬가지로 Weight는 5로 설정했으며, 버퍼 크기에 따라 EF 트래픽은 패킷 손실이 전혀 발생하지 않는 반면



(a) BE 트래픽



(b) EF 트래픽

그림 17. 버스티 입력 트래픽에 대한 패킷 손실

BE 트래픽의 패킷 손실은 상대적으로 높다. 그러나 버퍼 크기가 증가할수록 패킷 손실이 낮아짐을 알 수 있다.

그림 18은 EF와 BE 트래픽의 패킷 손실 정도를 한 그래프 내에 나타내었다. 이 그림에서는 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.

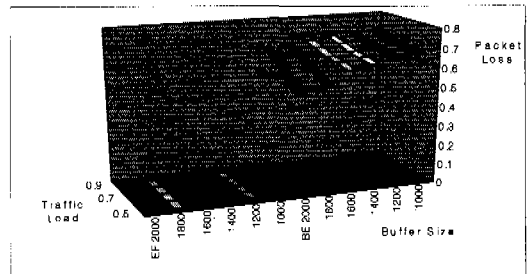
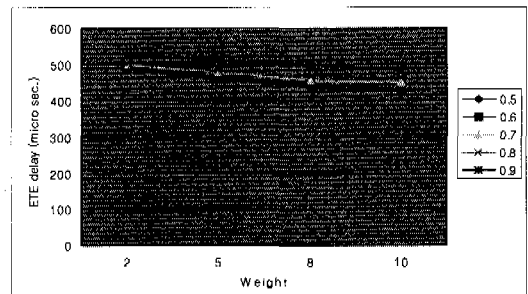


그림 18. 버스티 입력 트래픽에 대한 EF와 BE 클래스의 패킷 손실 비교

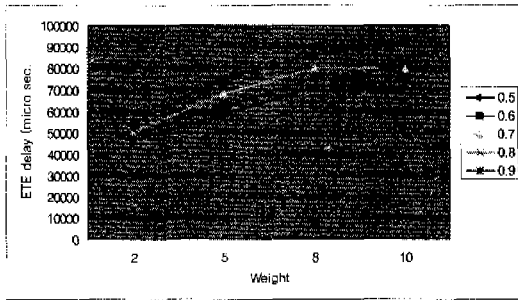
3.3 자기유사 트래픽에 대한 결과 및 분석

1) 평균 종단간 지연

그림 19의 (a)와 (b)는 자기유사 입력 트래픽 모델을 사용하였을 때, 인가 부하량에 대하여 EF와 BE 클래스의 평균 종단간 지연을 보여주고 있다. 자기유사 트래픽에 대해서도 위의 트래픽들과 마찬가지로 Weight가 높을수록 EF 트래픽의 평균 종단간 지연이 감소하는 반면, BE 트래픽의 평균 종단간 지연은 증가하는 것을 볼 수 있다. 그러나 랜덤 트래픽과 버스티 트래픽의 종단간 지연보다는 EF와 BE 클래스의 지연 값 모두가 많이 높아진 것을 알 수 있다.



(a) EF Traffic



(b) BE Traffic

그림 19. 자기유사 트래픽에 대한 종단간 지연

그림 20은 EF와 BE 클래스의 평균 종단간 지연을 동시에 비교한 것이다. 현재 인터넷망의 트래픽 특성으로 알려진 자기유사 특성을 가진 트래픽에 대해서도 DiffServ가 적용될 경우, EF 클래스의 종단간 지연 특성도 만족될 것으로 분석된다.

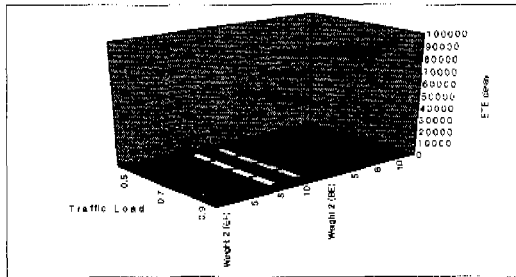
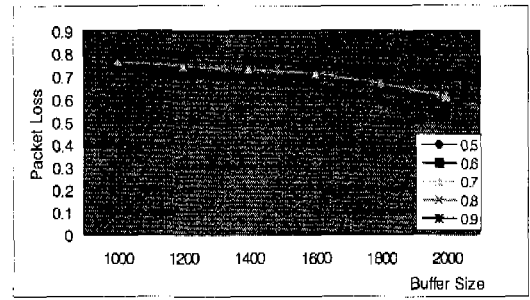


그림 20. 자기유사 입력 트래픽 모델에 대한 EF와 BE 클래스의 평균 종단간 지연 비교

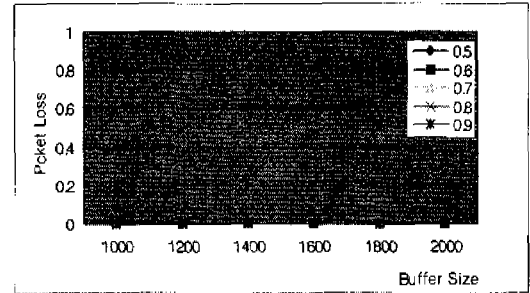
2) 패킷 손실

그림 21의 (a)와 (b)는 자기유사 입력 트래픽 모델을 사용하였을 때, 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. 그러나 랜덤 트래픽과 버스티 트래픽의 BE 트래픽 패킷 손실보다는 많이 높은 것을 알 수 있다. 역시 버퍼 크기가 증가할수록 손실은 감소되었다.

그림 22는 EF와 BE 클래스의 패킷 손실을 동시에 비교한 것이다. 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.



(a) BE Traffic



(b) EF Traffic

그림 21. 자기유사 트래픽에 대한 패킷 손실

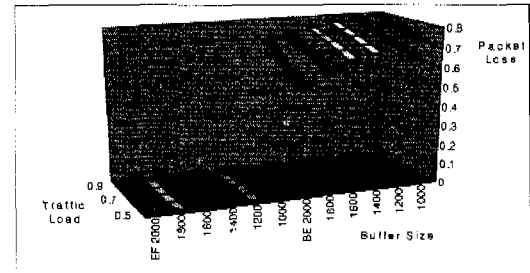
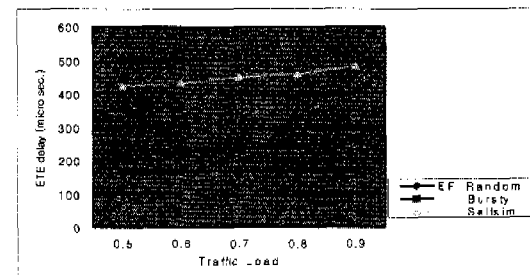


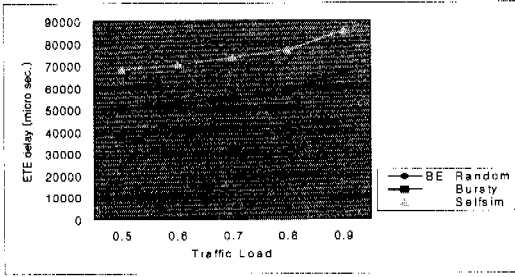
그림 22. EF와 BE 클래스의 패킷 손실 비교

3.4 트래픽 모델 별 성능 비교

그림 23 (a)와 (b)는 트래픽 모델에 따라 평균 종단간 지연 통계치로 EF와 BE 클래스의 성능을 분석한 결과를 나타내었다.



(a) EF Traffic



(b) BE Traffic

그림 23. 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 24는 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.

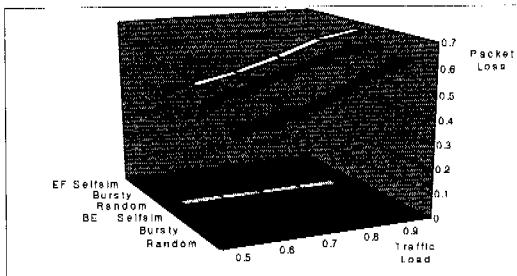


그림 24. 트래픽 특성에 따른 패킷 손실 비교

V. 역동적 행위 분석(Dynamic Behavior Analysis)

본 장에서는 차등 서비스의 EF와 BE 클래스간의 할당되는 Weight factor에 따른 서비스 메커니즘의 역동적 행위(dynamic behavior)를 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^[16]. 어떤 보존 법칙도 “플에게 값기 위하여 피터에게서 빌린다”는 원칙을 적용함으로써 직관적으로 묘사될 수 있다. 즉, 한 개이상의 트래픽 클래스를 가진 대기 시스템(queueing system)에서, 한 클래스의 트래픽에 주어지는 우선적인 취급은 성능 측정에 관계없이 다른 클래스에서의 비용으로 주어진다라는 것이다.

바로 이것이 차등 서비스에서 적용하는 기본적인 원칙이다. 즉, 똑 같이 주어지는 서비스가 아니라 사용자에게 따라서 서비스를 차별화하는 것이다. 적용되는 트래픽 스트림에 따라서 약간의 차이는 예상되지만, 차등 서비스 메커니즘의 시스템 행위의 기하학적 표현을 그림 25와 같이 기본적으로 나타낼 수 있다. 이 평면상에서, 작업 보존 정책(work-conserving policy)에 대한 성취할 수 있는 평균 지연의 집합은 선분 EG로 나타낼 수 있다. 여기서 점 E는 EF 클래스에 대한 최소 지연점을 보여주고, 점 G는 BE 클래스에 대한 최소 지연점을 나타낸다. 선 1은 지연에 대한 목표를 나타낸다. 세 개의 선 중에서, 선 1이 EF에 대하여 가장 우선 순위를 많이 준 것이고, 선 3(45도 각도)은 EF와 BE 클래스에 대하여 서비스 차등을 두지 않은 최선-노력 서비스에 해당한다.

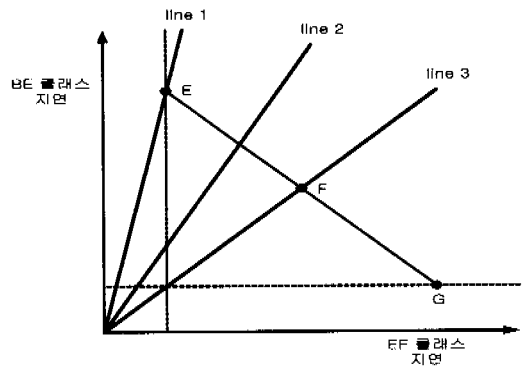


그림 25. 시스템 행위의 기하학적 표현

그림 26은 Weight 값 1, 3, 5, 8, 10으로 할당했을 때의 EF와 BE 트래픽에 대한 종단간 지연을 모의실험을 통하여 얻은 것을 보여준다. 이 그림에서도 EF 클래스에 weight 값을 많이 줄수록 EF 클래스의 지연은 감소하는 반면(즉, 그림 25에서의 점 E에 근접), BE 클래스의 지연은 증가되는 것을 보여준다. Weight 값이 1인 경우(즉, 최선-노력 서비스

에 해당), EF와 BE 클래스의 지연은 그림 25의 F 점에 해당하는 지역에 집중되어 있음을 확인 할 수 있다(그림 26에서는 표현 상 수직축의 값이 크게 나타나기 때문에 45도 각도에 나타나지 않음).

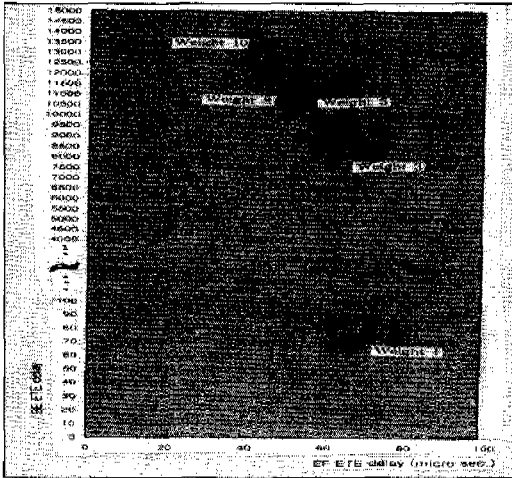


그림 26. Weight Factor에 의한 EF 및 BE 종단간 지연의 기하 분포

VI. 결론

본 논문에서는 트래픽 모델과 Weight Factor에 의한 DiffServ 서비스 메커니즘의 성능평가를 위하여 인터넷 트래픽을 특성화할 수 있는 트래픽 모델링에 대한 연구를 기반으로, DiffServ 시뮬레이터를 설계하고 구현하여 DiffServ의 QoS 보장 기술에 대한 성능 평가를 수행하였다. 이를 위하여 먼저 트래픽 모델을 크게 비자기유사 모델과 자기유사모델로 구분하였으며, 비자기유사 트래픽 모델로는 랜덤 트래픽 모델과 IBP 트래픽 모델을 사용하였다. 트래픽 모델의 새로운 패러다임으로 등장한 자기유사 트래픽 모델에 대하여도 연구를 수행하였으며, 중첩된 On/Off 프로세스를 이용한 비자기유사 트래픽 발생기를 설계하고 시뮬레이션에 사용하였다.

차등 서비스 메커니즘에 대하여는 제안 및 표준화된 DiffServ의 PHB 그룹들과, QoS를 보장할 수 있는 DS 네트워크 및 DS 라우터의 구성요소에 대해서 분석하였다. 이와 같은 DiffServ QoS 보장 기술에 대한 연구 내용을 바탕으로, 인터넷 QoS를 지원하기 위한 DiffServ 네트워크를 OPNET으로 구현해 성능평가를 하기 위한 방안을 제안하고 성능 분석을 수행하였다. 설계된 DiffServ 네트워크에서는 패킷 생성기를 이용하여 패킷을 발생시킨 후,

DiffServ 라우터에서 분류, 미터링 및 마킹을 하고 마킹 결과에 따라 WFQ 트래픽 스케줄링 정책으로 QoS를 제공받게 된다.

본 논문에서는 개발된 DiffServ 시뮬레이터를 통하여 앞에서 기술한 랜덤, 버스티, 자기유사 입력 트래픽 모델을 사용하였을 경우에 대하여, 다양한 입력 파라미터에 대하여 모의실험을 수행하고 성능 평가를 수행하였다. 성능분석 결과, 모든 환경에서 DiffServ 기능을 이용하여 EF와 BE 클래스의 서비스 분리가 가능함을 확인할 수 있었다. 그러나 BE 클래스에 대하여는 상황에 따라서 지연이 과도하게 발생함으로 인터넷망의 입력 단에서 수락 제어 등을 통한 트래픽 제어가 필요한 것으로 분석된다. 그리고 주어진 자원을 EF와 BE 클래스 사이에 효율적으로 분배하기 위하여 WFQ에서의 Weight factor에 대하여도 망의 상태를 고려한 튜닝 작업이 절대적으로 필요한 것으로 분석된다. 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.

DiffServ 네트워크에서 QoS를 제공하더라도 문제가 발생하는 경우는 폭주가 발생했을 때이다. 폭주를 미리 예방하거나 폭주를 빨리 벗어나는 것이 네트워크 성능에 중요하다. 그러므로 DiffServ 네트워크에 있는 라우터에 수락 제어 알고리즘을 반영하고, 버퍼관리 정책으로 RED나 WRED 기법을 적용시켜 네트워크 성능을 향상시키는 방안을 분석하고자 한다. 이것은 향후 연구 계획으로 남겨둔다. 또한 단일 DiffServ 영역이 아닌 복수의 영역으로 망을 확장시킨 DiffServ 네트워크를 구축하는 것과 이들 네트워크간에 대역폭 브로커(Bandwidth Broker: BB)를 사용해 SLA(Service Level Agreement)를 자동으로 협상하게 하는 것도 향후 연구 과제이다.

참고 문헌

- [1] S. Shenker, et al., Integrated Services in the Internet Architecture: an Overview. RFC 1633, Jun. 1994.

[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] K. Nichols, et al., A Two-bit Differentiated Services Architecture for the Internet, RFC 2638, July 1999.

[5] William Stallings, High-Speed Networks: TCP/IP and ATM Design Principles, Prentice-Hall, 1998.

[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] K. Nichols, et al., "Definition of the Differentiated Services Field(DS Field) in the IPv4 and IPv6 Headers", RFC 2474, Dec. 1998.

[10] Loukola M.V., et al., "Dynamic RT/NRT PHB group", Internet Draft<draft-loukola- dynamic-00.txt>, Nov. 1998.

[11] 전용희, "ATM 통신망에서의 지연 및 손실 우선 순위를 갖는 다중화 알고리즘의 성능평가", 한국통신학회논문지 제19권 5호, pp.842-856, 1994년 5월.

[12] I. Norros, "A storage model with self-similar input", Queuing System, 16:387-396, 1994.

[13] P., Ramakrishnan, "Self-similar traffic Models", CSHCN Tech. Report. 99-5, 1999.

[14] A. Erramilli et al., "Chaotic maps as models of packet traffic", ITC-14, 1994.

[15] Y. Bernet, et al., "A Conceptual Model for Diffserv Routers", Internet Draft, June, 1999.

[16] E. Gelenbe and I. Mitrani, Analysis and Synthesis of Computer Systems, New York, Academic Press, 1980.

전용희(Yong-Hee Jeon)

중신회원



1978년: 고려대학교 전기공학과 졸업(BS)

1989년: 미국 노스캐롤라이나 주립대 대학원 Elec. and Comp. Eng. 졸업(MS)

1992년: 미국 노스캐롤라이나 주립대 대학원 Elec. and Comp. Eng. 졸업(Ph. D.)

1978년~1978년: 삼성중공업(주) 근무

1989년~1989년: 미국 노스캐롤라이나주립대 Dept of Elec. and Comp. Eng. TA.

1989년~1989년: 미국 노스캐롤라이나주립대 부설 CCSP(Central For Comm. & Signal Processing) RA.

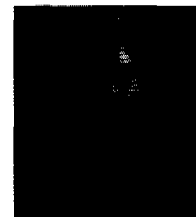
1992년~1994년: 한국전자통신연구원 교환전송 기술연구소 선임연구원

1994년~현재: 대구가톨릭대학교 공과대학 컴퓨터정보통신공학부 학부장, 공과대학장

<주관심 분야> 차세대 인터넷, 초고속 통신망 프로토콜, 통신망 성능분석, QoS 보장기술, 고속 통신망 응용 서비스, 통신망 보안

박수영(Soo-Young Park)

준회원



1991년: 대구가톨릭대학교 전산통계학과 학사

1996년: 대구가톨릭대학교 대학원 전산통계학과 석사

1999년: 대구가톨릭대학교 대학원 전산통계학과 박사과정 수료

1990년~1992년: 대구백화점(주)정보시스템부 근무

1996년~1999년: 대구미래대학 멀티미디어정보과학과 겸임교수

2001년 3월~현재: 위덕대학교 컴퓨터공학과 전임강사

<주관심 분야> 차세대 인터넷, QoS 보장기술, 고속 통신망 응용 서비스, 통신망 보안