
패킷 크기에 따른 처리율 영향을 완화하는 큐 관리 기법 연구

이 성 근*

A Study on a Queue Management Scheme for Alleviating the Impact of Packet Size on the Throughput

Sung-Keun Lee*

이 논문은 정보통신부에서 지원하는 대학기초연구지원사업으로 수행하였음.

요 약

차별 서비스 모델은 IP 기반 네트워크에서 서비스 품질을 제공하기 위한 새로운 구조 모델이다. RIO 는 차별 서비스 네트워크 환경에서 라우터의 큐 관리 메커니즘으로 가장 널리 적용되고 있다. 본 논문은 패킷 길이에 따른 처리율의 영향을 완화하는 개선된 RIO 메커니즘을 제안한다. 트래픽 조절 기능인 TSW 기능과 결합하여 시뮬레이션을 수행한 결과, 제안된 메커니즘은 적절히 배치된 네트워크 상황에서 패킷 길이에 무관하게 안정된 처리율을 보장하고, 대역폭의 공평한 분배를 제공하는 것으로 분석된다.

ABSTRACT

Differentiated services (DiffServ) architecture defines a new framework for the support of quality of service (QoS) in IP-based networks. RIO has received the most attention among all the active queue management mechanisms proposed for assured service in DiffServ environment. The paper proposed the enhanced RIO which could alleviate the impact of flow's packet size on the realized throughput. The simulation results indicate that this mechanism, when combined with TSW as traffic conditioner, provide better throughput assurance and fair distribution of excess bandwidth independent of packet size of flows in case of well-provisioned network environment.

키워드

Differentiated Services Network, Queue Management, RIO, Assurance services

1. 서 론

차별 서비스 네트워크는 인터넷을 기반으로 실시간 멀티미디어 서비스를 제공하기 위해 서비스에 따라 차별화된 QoS(Quality of Service)를 보장하기 위한 개념이다. 차별 서비스는 자원 예약 및 각 흐름

에 대한 상태 관리를 필요로 하는 통합 서비스 모델^[1]과 달리, 트래픽을 흐름별로 다루지 않고 서비스 클래스로 분류하여 처리함으로써 중간 라우터에서 고속의 패킷 전달을 할 수 있도록 한다^[2]. 차별 서비스 네트워크의 구성 요소는 그림 1과 같다^{[3][4]}.

* 순천대학교 컴퓨터교육과

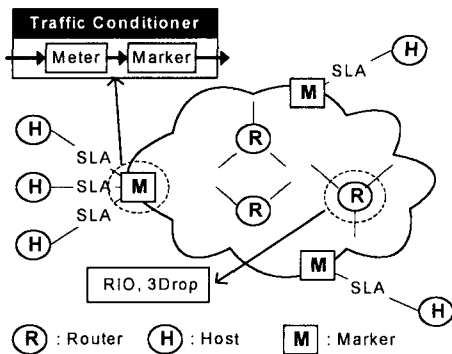


그림 1. 차별 서비스 네트워크
Fig 1. Differentiated services network

트래픽 조절 기능은 흐름 또는 흐름 그룹별로 유입되는 전송률을 측정하여, 사전에 약속된 계약 전송률의 준수 여부에 따라 패킷 헤더에 IN 또는 OUT으로 마킹하여 패킷을 네트워크로 전달한다. 차별 서비스 영역내의 라우터는 패킷헤더 정보(IN 또는 OUT)에 따라 차별적으로 패킷을 전달한다^{[3][5]}. RIO(RED with IN/OUT)는 차별 서비스를 실현하기 위해 적용되는 가장 기본적이고 대표적인 큐 관리 기법이다^[2]. 차별 서비스 모델은 RIO와 트래픽 조절 기능을 결합하여, 네트워크 혼잡 상황에서도 최소 대역폭을 보장할 수 있다는 연구 결과가 나타났다^{[3][4]}. 그러나 RTT(Round Trip Timer), 예약된 대역폭의 크기, UDP의 영향 및 결합되는 흐름의 개수에 따라 서비스 보장이 달라지는 문제점이 있다^{[7][8]}. 또한 차별 서비스 영역으로 유입되는 흐름들의 패킷 길이에 따라 처리율이 크게 차이나는 문제점이 지적되었다^[6]. 즉, 동일한 예약 대역폭을 갖는 흐름 중에서 패킷의 길이가 긴 흐름이 더 많은 대역폭을 차지하게 된다. 현재의 인터넷은 서로 상이한 여러 개의 네트워크로 결합되고, 네트워크마다 MTU(Maximum Transfer Unit)가 다를 수 있다. 또한 동일한 TCP 소스에서 생성된 TCP 세그먼트가 여러 개의 IP 네트워크를 통해 전달될 때 MTU에 따라 분할되어 전달되는 상황이 발생할 수 있다. 따라서 패킷 길이에 따른 처리율의 편차는 대역폭의 불공평한 분배를 초래하므로 반드시 해결되어야 한다. 본 논문은 패킷의 길이에 따른 처리율의 편향을 완화시키는 큐 관리 기법을 제안한다. 제

안한 알고리즘은 RIO에 기반을 두고, 패킷의 길이에 따라 OUT 패킷 폐기 확률을 차등적으로 적용한다. 제안한 알고리즘의 성능 분석은 ns-2를 이용한 시뮬레이션 방식에 의해 수행한다. 본 논문의 구성은 다음과 같다.

제2장에서는 RIO의 기능에 대해 설명하고, 패킷 길이에 대한 문제점을 시뮬레이션을 통해 제기한다. 제3장에서는 RIO를 확장한 큐 관리 기법에 대해 제안하고, 제4장에서는 시뮬레이션을 통해 제안한 기법의 성능을 분석한 후, 제5장에서 결론을 맺는다.

II. 차별 서비스 모델의 큐 관리 기능

차별 서비스 도메인의 내부 라우터에서 적용되는 큐 관리 기능은 입력되는 패킷의 코드값과 큐의 현재 패킷 수를 기반으로 패킷을 전달 또는 폐기한다. 본 절에는 대표적인 큐 관리 기법인 RED(Random Early Detect), RIO의 기능에 대해 설명하고, 시뮬레이션을 통해 RIO의 패킷 길이에 따른 처리율의 편향성에 대해 분석한다.

2.1 RED 큐 관리 기법

RED는 측정된 평균 큐 길이를 기초로 패킷 폐기 확률을 결정함으로써 패킷을 공평하고, 안정적으로 폐기하도록 한다^[9]. RED는 혼잡 상황이 발생한 후 이를 해결하기 위한 방법이 아니라 혼잡 상황을 사전에 방지하는 개념이다. 즉, RED는 큐가 완전히 차지 않더라도 어떤 한계치(Min_{th})를 넘으면 일정한 확률을 가지고 미리 패킷을 폐기함으로써 큐가 넘치는 현상을 미연에 방지한다. Drop tail 방식은 망 혼잡시, 유입되는 패킷을 지속적으로 폐기하므로 다수의 TCP 흐름이 동시에 전송률을 감소시키고, 일정시간 후에는 다시 동시에 전송률을 증가시켜서 네트워크를 불안정하게 하여 심각한 성능저하를 초래한다. RED는 무작위로 패킷을 폐기시킴으로써 각 종단 시스템에서 TCP 혼잡 제어 기능을 서로 다른 시간에 이루어지도록 하여 동기화 문제를 해결할 수 있다.

2.2 RIO 큐 관리 기법

차별 서비스 네트워크의 큐 관리 기법으로 RED를 기반으로 한 RIO를 제안하였다^[3]. RIO는 IN 패킷과 OUT 패킷 각각에 대한 RED 알고리즘을 사용하며, 각 큐에는 두개의 임계값(Min_{th} , Max_{th})을 둔다. 입력되는 패킷이 IN이면, 큐 내에 존재하는 IN 패킷의 개수를 측정하여, 패킷의 개수가 $Min_{in,th}$ 이하이면, 입력된 패킷을 전달한다. 두 임계값 사이에 있으면 패킷을 확률적으로 폐기한다. 망 혼잡이 발생하여 큐 길이가 $Max_{in,th}$ 이상이면 모든 패킷을 무조건 폐기한다. 입력되는 패킷이 OUT이면, 큐 내에 존재하는 전체 패킷의 개수를 측정하여, 패킷 전달 여부를 결정한다.

RIO는 그림 2에 나타낸 바와 같이 IN, OUT에 대한 임계값 및 패킷 폐기 확률을 차별적으로 적용한다. 즉, 네트워크 부하가 일정한 한계치를 넘어서면 OUT 패킷을 우선적으로 폐기시킴으로써, 망 혼잡 발생시에도 계약된 전송률로 데이터를 전송하는 가입자에게 최소한의 예측 가능한 서비스를 제공할 수 있다. 또한 네트워크 부하가 낮을 시에는 더 많은 OUT 패킷이 전송되므로 초과 대역폭을 공유하며, 망의 효율을 증가시킨다.

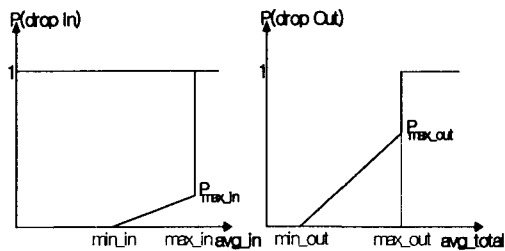


그림 2. RIO 알고리즘의 폐기 확률
Fig 2. Drop probabilities of RIO algorithm

2.3 패킷 길이에 따른 처리율의 영향 분석

본 절에서는 RIO를 큐 관리 기법으로 적용했을 때 각 흐름의 패킷 길이에 따라 처리율의 편차가 나타나는 현상을 분석하기 위해 시뮬레이션을 수행한다.

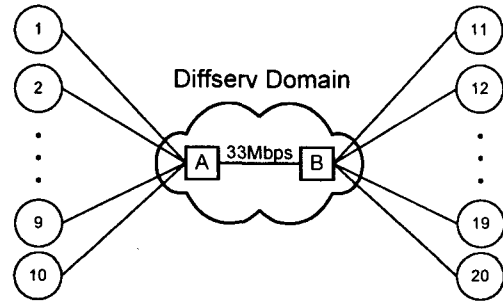


그림 3. 시뮬레이션을 위한 네트워크 구성도
Fig 3. Network topology Simulation

시뮬레이션 도구는 ns-2^[10]를 이용하였고, 네트워크 구조는 그림 3과 같이 라우터 A와 B 사이에 하나의 공통 링크로 연결되며, 백본 링크의 대역폭은 33 Mbps를 갖도록 하였다. 각 라우터에는 10개의 호스트가 연결되어 있다. 각 호스트 (i)는 상대 호스트 (i+10)과 TCP 연결을 설정한다. 모든 흐름의 RTT는 20msec로 설정하였다. 경계 라우터의 트래픽 조절기능은 TSW(Time Sliding Window)를 적용하였고^[3], 기본 윈도우 크기는 600msec로 지정하였다.

시뮬레이션은 30초 동안 실행하여, TCP가 안정 상태에 도달한 마지막 20초 동안 싱크에서 수신한 패킷의 처리율을 측정한다. 동일한 환경에서 seed를 달리하면서 7번 시뮬레이션 하여 최대, 최소값을 제외한 결과의 평균값을 취하였다. TCP 버전은 현재 가장 많이 적용되는 TCP Reno를 이용하였고, 트래픽 종류는 ftp 트래픽으로 가정하였다. 시뮬레이션에 적용된 RIO 파라미터를 표 1에 나타내었다.

표 1. RIO 파라미터 값
Table 1. The values of RIO parameter

파라미터	설정값
$Min_{out,th}$	10
$Max_{out,th}$	30
P_{out}	0.2
$Min_{in,th}$	40
$Max_{in,th}$	70
P_{in}	0.02

첫 번째 시뮬레이션은 모든 흐름이 동일한 예약 전송률을 갖는다고 가정하였다. 즉, 10개의 호스트가 모두 3Mbps의 예약 전송률을 갖고, 5개의 그룹이 각각 512, 1024, 1536, 2048, 2560 바이트의 패킷 길이를 가질 때 각 흐름별 처리율을 표 2에 나타내었다.

표 2. 각 흐름별 처리율(case1)
Table 2. Realized throughput per flow(case1)

flow	패킷길이 (byte)	목표 전송률 (Mbps)	처리율 (Mbps)
1	512	3.3	2.71
2	512	3.3	2.76
3	1024	3.3	3.07
4	1024	3.3	3.00
5	1536	3.3	3.05
6	1536	3.3	3.35
7	2048	3.3	3.20
8	2048	3.3	3.50
9	2560	3.3	3.32
10	2560	3.3	3.55

이상적인 경우 모든 흐름들이 동일한 처리율을 달성하여야 하는데, 표 2의 결과는 패킷 길이에 따라 처리율의 편차가 뚜렷이 나타난다. 즉, 예약 전송률이 같더라도 패킷의 길이가 긴 흐름(흐름 10)이 패킷 길이가 상대적으로 짧은 흐름(흐름 1)보다 더 많은 처리율을 달성하였다. 이러한 결과는 TCP 흐름의 성능 모델인 식 (1)과 부합한다^[11].

$$T \leq 1.5 \cdot \sqrt{2/3} \cdot B / (R \cdot \sqrt{p}) \quad (1)$$

식(1)에서 T는 TCP의 기대되는 최대 처리율을 나타내고, B는 패킷의 길이, R은 round trip time, p는 패킷의 손실 확률을 의미한다. 표 2의 결과는^[6]의 연구 결과와도 일치한다.

두 번째 시뮬레이션은 두 개의 그룹이 서로 다른 예약 전송률을 갖는 상황을 가정하여 수행하였다. 10개의 소스를 두 개의 그룹으로 구분하여, 첫 번째 그룹(흐름 1~5)은 1Mbps의 예약 대역폭을 지정하고, 두 번째 그룹(흐름 6~10)은 5Mbps의 대역폭을 할당하였다. 동일 그룹 내에서 각각의 흐름의

패킷 길이를 512, 1024, 1536, 2048, 2560 바이트로 지정하여 시뮬레이션을 수행하였다. 각 흐름별 처리율을 표 3에 나타내었다.

표 3. 각 흐름별 처리율(case 2)
Table 3. Realized throughput per flow(case 2)

flow	패킷길이 (byte)	예약 전송률 (Mbps)	처리율 (Mbps)
1	512	1	1.04
2	1024	1	1.34
3	1536	1	1.62
4	2048	1	1.82
5	2560	1	1.94
6	512	5	4.23
7	1024	5	4.65
8	1536	5	5.12
9	2048	5	4.99
10	2560	5	5.01

표 3의 결과를 분석하면, 첫 번째 그룹은 패킷 길이에 따라 처리율의 편차가 확연히 나타나는데 비해, 두 번째 그룹은 약간 불규칙적인 추이를 나타내지만 패킷 길이에 대한 편차가 여전히 존재한다.

따라서 모든 흐름이 동일한 예약 전송률을 갖는 상황뿐만 아니라 다양한 예약 전송률이 결합된 상황에서도 패킷 길이에 따른 처리율의 차이가 발생함을 알 수 있다.

III. 개선된 큐 관리 기법 제안

RIO-TSW에 기반한 차별 서비스 네트워크는 동일한 예약 대역폭을 갖더라도 각 소스에서 생성되는 패킷 길이에 따라 각 흐름이 달성하는 처리율이 달라짐을 알 수 있다. 현재의 인터넷은 이질적인 여러 네트워크로 결합되고, 각 네트워크마다 최대 전송할 수 있는 패킷의 길이가 다를 수 있다. 또한 동일한 TCP 소스에서 생성된 TCP 세그먼트가 여러 개의 IP 네트워크를 통해 전달될 때 MTU에 따라 분할되어 전달되는 상황이 자주 발생할 수 있다. 따라서 패킷 길이에 따른 처리율의 편차는 대역폭의

불공평한 분배를 초래하므로 해결하여야 한다.

TSW는 유입되는 패킷의 전송률을 바이트 단위로 측정하여, 계약률의 초과 여부에 따라 IN 또는 OUT으로 마킹한다^[3]. 따라서 TSW는 패킷 길이에 대한 차이에 대해 적절히 대응한 것으로 판단된다. 이러한 추론은 표 4의 결과를 통해 확인할 수 있다. 표 4는 첫 번째 시뮬레이션 수행 동안 각 흐름별로 TSW에서 마킹한 IN 패킷과 OUT 패킷의 개수를 나타내었다.

표 4의 결과를 분석하면 각 흐름별로 IN 패킷의 개수는 패킷 길이에 따라 일정한 비율로 반비례하여 전달된다. 즉, 흐름 1이 흐름 10에 비해 패킷의 길이가 1/5이므로, 동일한 예약 전송률을 성취하기 위해서는 5배의 패킷 수를 전달하여야 하는데 표 4의 결과는 근사적으로 일치한다. 그러나 OUT 패킷의 개수는 패킷 길이와 관계없이 각 흐름이 비슷한 양을 생성한다.

따라서 각 흐름의 수신측에 전달되는 OUT 패킷의 개수에 따라 처리율의 차이가 발생함을 알 수 있다.

표 4. 각 흐름별 전송한 패킷 수
Table 4. Packet numbers sent per flow

flow	IN 패킷수	OUT 패킷수	전체 패킷수
1	12065	1225	13,291
2	12927	1432	14,359
3	6339	1609	7948
4	6067	1607	7674
5	4073	1395	5468
6	3755	1184	4939
7	2868	1134	4002
8	2928	1189	4117
9	2332	1011	3343
10	2398	1090	3488

본 논문에서는 RIO에서 OUT 패킷에 대한 폐기 확률을 계산할 때, 큐 내에 존재하는 전체 패킷의 평균 개수 뿐만 아니라 현재 처리되는 패킷의 길이를 반영하여 패킷 폐기 확률을 결정하도록 제안한다. 패킷 길이가 큰 흐름으로부터 생성된 OUT 패킷의 폐기 확률을 상대적으로 증가시킴으로서, 패킷의 길이와 무관하게 모든 흐름이 자신의 예약 전송

률에 따라 처리율을 달성하도록 한다. 그림 4는 기존 RIO를 확장한 알고리즘을 나타내었다.

IN으로 마킹된 패킷을 수신했을 때의 처리과정은 RIO 알고리즘과 동일하다. OUT 패킷을 수신했을 때는 먼저, 패킷의 길이를 구한 후, 평균 패킷 길이를 의미하는 상수 k를 통해 size factor를 결정한다. OUT 패킷을 연속적으로 폐기하지 않고 일정한 간격으로 폐기하기 위한 상태변수 count를 size factor를 고려하여 증가시킨다. 또한 OUT 패킷의 폐기 확률을 결정할 때 기존 RIO에서는 avg_{total} 만을 고려하였으나, 본 방법은 입력된 패킷과 평균 패킷 길이의 관계(sizeFactor)를 고려하여 폐기 확률 결정한다.

따라서 평균 패킷 길이를 초과하는 흐름에 대해서 OUT 패킷의 폐기 확률을 증가시키고, 평균 길이에 미치지 못한 패킷에 대해서는 폐기 확률을 감소시킴으로서 패킷 길이와 무관하게 자신의 예약 대역폭을 달성하도록 한다.

```

for each arriving packet P
    calculate  $avg_{in}$ 
    calculate  $avg_{total}$ 
    if P is marked as IN
        if  $min_{th_{in}} < avg_{in} < max_{th_{in}}$ 
            calculate  $P_{in}$ 
            drop P with prob.  $P_{in}$ 
        else if  $max_{th_{in}} < avg_{in}$ 
            drop P
        else
            enqueue P
    else if P is marked as OUT
        if  $min_{th_{out}} < avg_{total} < max_{th_{out}}$ 
            sizeFactor = size(P) / k;
            if (sizeFactor > 1.0) count += sizeFactor;
            else count++;

             $P_b = P_{max_{out}} * (avg_{total} - min_{th_{out}}) / (max_{th_{out}} - min_{th_{out}})$ 
             $P_a = P_b / (1.0 - count * P_b)$ ;

            if (sizeFactor <= 1.0)
                 $P_{out} = P_a * (27 * (sizeFactor - 0.25) / 75 + 0.73)$ ;
            else
                 $P_{out} = P_a * ((sizeFactor - 1.0) / 5 + 1.0)$ ;

            drop P with prob.  $P_{out}$ 
        else if  $max_{th_{out}} < avg_{total}$ 
            drop P
        else
            enqueue P
    
```

그림 4. 개선된 RIO 알고리즘의 코드
Fig 4. Pseudo code of enhanced-RIO algorithm

IV. 시뮬레이션 및 분석

제안된 큐 관리 기법의 성능 분석을 위해 ns-2의 RIO 알고리즘을 확장하여 구현하였고, 제2장에서 수행한 네트워크 구조 및 파라미터를 그대로 적용하여 시뮬레이션을 수행하였다.

표 5는 모든 흐름의 예약 전송률이 3Mbps 인 경우의 RIO와 개선된 RIO를 적용했을 때의 각 흐름의 처리율을 비교 분석한 결과이다. 각 흐름이 패킷 길이에 관계없이 목표 대역폭에 근접한 처리율을 성취함을 알 수 있다.

그림 5는 표 5의 결과를 그래프로 나타낸 것으로 개선된 RIO가 공평한 대역폭 분배를 실현시킴을 보여준다.

표 5. RIO와 개선된 RIO 간의 비교(case1)
Table 5. Comparison of RIO and E-RIO(case1)

flow	패킷 길이 (byte)	처리율 (Mbps)	
		RIO	Enhanced RIO
1	512	2.71	3.10
2	512	2.76	2.97
3	1024	3.07	3.13
4	1024	3.00	3.14
5	1536	3.05	3.10
6	1536	3.35	3.14
7	2048	3.20	3.16
8	2048	3.50	2.97
9	2560	3.32	3.22
10	2560	3.55	3.14

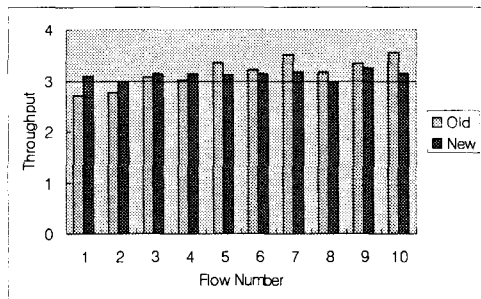


그림 5. RIO와 개선된 RIO 간의 비교도
Fig 5. Comparison of RIO and E-RIO

표 6은 두 번째 시뮬레이션 결과를 통해 RIO와 개선된 RIO 간의 차이점을 비교하였다.

표 6의 결과를 분석하면 예약 대역폭이 1Mbps인 그룹과 5Mbps인 그룹 모두에서 개선된 RIO가 처리율의 균등한 분배를 실현시킴을 나타낸다. 그러나 두 번째 그룹에서 패킷 길이에 대한 편차가 약간 불규칙적인 형태로 나타난다. 이것은 [3][7][8]에서 지적하였듯이, 예약 전송률의 차이에 따른 처리율의 편향현상에 기인하여 그룹 1의 대부분의 흐름들이 자신들의 목표 전송률을 초과 달성하게 된다. 따라서 그룹 2 흐름의 처리율 손실이 발생하게 되어, 그룹 1에 비해 불규칙적인 편차가 나타난 것으로 분석된다. 따라서 다양한 예약 전송률이 결합된 상황에서도 개선된 RIO는 패킷 길이에 따른 처리율의 차이를 크게 개선시킨다고 할 수 있다.

표 6. RIO와 개선된 RIO 간의 비교(case2)
Table 6. Comparison of RIO and E-RIO(case2)

flow	패킷 길이 (byte)	처리율 (Mbps)	
		RIO	Enhanced RIO
0	512	1.04	1.53
1	1024	1.34	1.50
2	1536	1.62	1.59
3	2048	1.82	1.60
4	2560	1.94	1.55
5	512	4.23	4.76
6	1024	4.65	4.71
7	1536	5.12	4.43
8	2048	4.99	4.89
9	2560	5.1	4.98

V. 결론

차별 서비스 네트워크 모델은 동일한 QoS를 요구하는 흐름들을 인공한 집합으로 분류하여, 각 집합에 대해 서비스를 차별적으로 제공한다. 따라서 차별 서비스 모델에서 성취해야 할 두 가지 필수조건은 각 흐름은 평균적으로 사전 계약된 만킬 데이터 전달을 수행해야 한다는 전송률 보장문제와 인

여 대역폭의 공평한 분배 문제를 들 수 있다. 그러나 RIO-TSW를 적용하였을 때 각 흐름별 생성되는 패킷의 길이에 따라 처리율이 크게 달라지는 문제가 시뮬레이션을 통해 지적되었다. 본 논문에서는 패킷의 길이에 따라 처리율의 편차를 완화시키는 큐 관리 기법을 제안하였다. 제안한 방법은 RIO 에 기반을 두고, 기존의 큐 내의 패킷의 점유율 뿐 만 아니라 패킷의 길이에 따라 OUT 패킷의 폐기 확률을 차등적으로 적용하였으며, 성능 분석은 ns-2를 이용한 시뮬레이션 방식에 의해 수행하였다.

본 논문에서 제안한 방법은 각 소스에서 발생하는 패킷 길이에 무관하게 예약 대역폭에 근접한 처리율을 달성하는 것으로 나타났다. 향후 count 상태 변수 값의 변경식 및 OUT 패킷을 폐기하는 확률 함수에 대한 세부적 분석이 필요하며, 다양한 네트워크 구조 및 트래픽 소스가 포함된 환경에 대한 연구가 수행되어야 할 것이다.

참 고 문 헌

[1] R. Braden, et al., Integrated Services in the Internet Architecture: an Overview, Internet RFC 1633, July 1994.

[2] S. Blake, et al., An Architecture for Differentiated Services, Internet RFC 2475, Dec. 1998.

[3] D. Clark, W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service", ACM Transactions on Networking, Aug., 1998.

[4] I. Yeom and N. Reddy, Impact of marking strategy on aggregated flows in a DiffServ network, IEEE ICMCS '99, 1999.

[5] Y. Bernet., et al., A Conceptual Model for DiffServ Routers, IETF Internet Draft, draft-ietf-diffserv-model-02.txt, March 2000.

[6] N. Seddigh, et al., Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network, Globecom '99, Dec., 1999

[7] W. Lin, R.Zheng and J. Hou, How to Make Assured Services More Assured, ICNP '99, Oct. 1999.

[8] I. Yeom, J. Hou, Realizing Throughput Guarantees in a Differentiated Services Network, ICMCS, Florence, Italy, June 1999.

[9] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, vol.1 n.4, Aug. 1993

[10] Network simulator (Ns), University of California at Berkeley, CA, 1997. Available via <http://www-nrg.ee.lbl.gov/ns/>.

[11] S. Floyd and K. Fall, "Router mechanisms to support end-to-end congestion control", LBL Technical Report, Feb. 1997.

저 자 소 개



이성근(Sung-Keun Lee)

1985년 고려대학교 전자공학과 (공학사)

1987년 고려대학교 대학원(공학 석사)

1995년 고려대학교 대학원(공학 박사)

1996년 삼성전자 네트워크개발팀(선임연구원)

1997년~현재 순천대학교 컴퓨터교육과 조교수

※ 관심분야 : 인터넷 QoS, 분산 멀티미디어, 광대역 통신 프로토콜