

차별서비스 네트워크에서 보장형 서비스의 공정성 향상을 위한 이중 윈도우 기반 마킹 알고리즘

조 병 규[†] · 이 성 근^{††} · 강 의 성^{†††}

요 약

최근 인터넷에서 QoS(Quality of Service)를 제공하는 방법인 차별 서비스에 대해 많은 연구가 진행중이다. 차별화 서비스에서 보장형 서비스는 2등급 패킷 패기 우선순위 방법과 RIO(RED with In and Out)를 일반적으로 사용한다. 하지만 이 방법만으로는 대역폭을 만족할 만큼 공평하게 사용자에게 제공하지 못한다. 본 논문에서는 공정성을 개선하기 위한 방법으로 TSW(Time Sliding Window) marking algorithm을 개선하여 TS2W3C(Time Sliding Two Window Three Color)marking algorithm을 제안한다. ns-2 툴 이용하여 제안된 메커니즘을 모의 실험을 통해 성능을 분석한 결과, 제안한 방법이 기존의 방법에 비해 공평하게 대역폭을 분배하는 것으로 나타났다.

The Two Window-Based Marking Algorithm For Enhancing Fairness of Assured Services in a Differentiated Services Network

Byeong-Kyu Cho[†] · Sung-Keun Lee^{††} · Eui-Sung Kang^{†††}

ABSTRACT

In recent research for the Internet, many studies have investigated the Diffserv AS architecture that can provide Quality of Service. However, this architecture still lacks the qualification to provide full use of the bandwidth to the customer. In this paper, we propose the TS2W3C (Time Sliding Two Window Three Color) marking algorithm to improve the fair share of bandwidth by enhancing the TSW (Time Sliding Window) marking algorithm. Our proposed mechanism provides the bandwidth relatively more fairly than the TSW mechanism.

키워드 : TSW, 3컬러 마킹(Three color marking), 보장형 서비스(Assured services), 차별서비스(Differentiated services)

1. 서 론

최근 인터넷에서 서비스 품질(Quality of Service : QoS)을 보장하는 방법중의 하나인 차별서비스에 대해 많은 연구가 진행중이다. 차별서비스 네트워크 모델은 동일한 서비스 품질을 요구하는 플로우들을 일정한 플로우의 집합(aggregate)으로 분류하여, 각 집합에 대해 서비스를 차별적으로 제공한다[1].

차별 서비스 네트워크에서는 하나의 플로우에 대하여 세 개의 서비스 클래스(프리미엄 서비스, 보장형 서비스, 최선형 서비스)중 하나의 서비스 클래스를 제공하는데 이때 프리미엄 서비스는 일정한 대역폭을 보장하여 네트워크 내에서 패킷의 전달에 대해 최소지연과 안정된 지연변이를 보

장하는 방식이다.

보장형 서비스는 폭주가 발생하여 패킷이 손실되거나, 네트워크에 부하가 증가하여 사용할 수 있는 대역폭이 낮을 때에도 고객에게 최소한의 처리율(throughput)을 보장해 주기 위해 고안되었다.

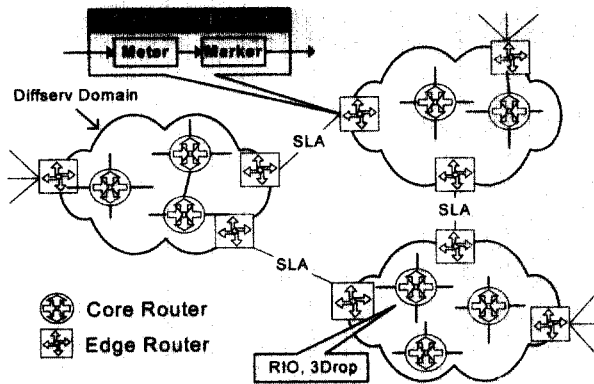
최선형 서비스는 대역폭을 모든 데이터의 전송에 균일하게 할당하는 기존의 최선형 서비스 방식이다.

현재 차별서비스의 보장형 서비스에서는 마커에서 패킷의 패기 우선순위를 IN과 OUT으로 구분하여 마킹을 하고, 내부 라우터의 큐 관리 알고리즘은 일반적으로 RIO(RED with In and Out)를 사용한다. 그러나 보장형 서비스의 품질은 RTT(Round Trip Time), 예약된 대역폭의 크기, 그리고 UDP source의 존재 유무에 의해 크게 좌우되는데, 이것은 처리율 보장 및 여분대역폭의 공정한 분배에 커다란 문제점으로 작용한다[2, 7].

차별 서비스 네트워크의 구조는 (그림 1)에 나타난 바와 같이 SLA(service level agreement), 경계 라우터의 트래픽

* 이 논문은 정보통신부에서 지원하는 2002년 정보통신기초기술연구지원사업으로 수행하였음.
† 준 회원 : 순천대학교 대학원 컴퓨터과학과
†† 정 회원 : 순천대학교 정보통신공학부 조교수
††† 정 회원 : 순천대학교 컴퓨터교육과 전임강사
논문접수 : 2002년 7월 29일, 심사완료 : 2002년 10월 1일

조절 기능과 내부 라우터의 큐 관리 기능으로 구성된다[3].



(그림 1) 차별 서비스 네트워크 구조

본 논문에서는 보장형 서비스에서 대역폭의 예약 크기에 따라 나타나는 처리율의 편향현상(bias)을 완화시킬 목적으로 TSW(time sliding window) 마킹 알고리즘[2]을 개선하여 TS2W3C(time sliding two window three color) 마킹 알고리즘을 제안한다. 제안한 메커니즘의 특징은 기존 TSW의 단점을 보완하기 위해 두개의 윈도우를 적용하여 하나는 짧은 시간동안의 전송률을 측정하고, 다른 하나는 상대적으로 긴 시간동안 전송률을 측정하게 한 것이다.

제안한 알고리즘의 성능분석은 ns-2 [4]를 이용한 시뮬레이션 방식에 의해 수행한다.

2. Time Sliding Window 마킹 알고리즘

TSW 마킹 알고리즘은 보장형 서비스를 위한 트래픽 조절기법 중 평균을 평가에 기반을 둔 방식으로서, 경계라우터의 서비스 프로파일에 명시된 목표 전송률 초과 여부를 기반으로 패킷 우선 순위를 결정하여 패킷을 IN이나 OUT으로 마킹 한다. TSW는 기본적으로 두 가지 독립적인 요소를 가지는데, 이들은 정해진 단위시간동안 일정하게 전송률을 측정하는 전송률 측정기와 이 전송률측정기에 의해 측정된 전송률을 기초로 패킷을 IN과 OUT으로 마킹하는 마킹 알고리즘이다.

TSW는 매 패킷이 도착할 때마다 전송속도를 측정하게 되며 윈도우의 크기에 따라 과거의 전송률에 대한 정보가 현재의 전송률 계산에 반영되는 정도가 달라지게 된다. TSW는 Win_length에 명시된 시간만큼 최근의 평균전송률을 저장하며, (그림 2)에서 알 수 있듯이 전송률 측정부분에서 현재의 전송률을 과거의 전송률과 합하여 평균을 내므로 TCP 트래픽의 버스트한 특성을 완만하게 만드는 작용을 하게 된다.

(그림 3)에서 보는 바와 같이 어느 순간의 전송률 R이 예약전송률 R_T 보다 작으면 패킷을 IN으로 마킹하고 그렇지 않은 경우는 확률적으로 패킷을 OUT으로 마킹하게 된다.

초기값:

$Win_length = a\ constant;$
 $Avg_rate = connection's\ target\ rate,\ R_T;$
 $T_front = 0;$

매패킷 도착시:

$Bytes_in_TSW = Avg_rate \times Win_length;$
 $New_bytes = Bytes_in_TSW + pkt_size;$
 $Avg_rate = New_bytes / (Now - T_front + Win_length);$
 $T_front = now;$

* now : 현재패킷이 도착한 시간
 pkt_size : 도착한 패킷의 크기

(그림 2) TSW 전송률 측정 알고리즘

측정된 전송률 R에 대하여

if $R < R_T$
 mark packet as **IN**.
 else
 calculate $P_{drop} = (R - R_T) / R_T;$
 with P_{drop} , mark packet as **OUT**.

(그림 3) TSW 마킹 알고리즘

TSW에서는 Win_length의 크기에 따라 전송률에 대한 정보가 다르게 해석될 수 있는데, Win_length가 상대적으로 짧을 경우 순간적으로 변하는 트래픽에 대한 정보를 정확하게 파악 할 수 있지만 긴 시간에 대한 정보를 가지지 못함으로 연결시간이 긴 TCP 트래픽의 경우 평균 전송률에 대한 정보의 신뢰성이 저하된다. 그리고 Win_length가 상대적으로 긴 경우에는 일정기간 동안의 평균전송률에 대한 정보는 비교적 정확한 값을 얻을 수 있으나, 순간적으로 변하는 버스트한 트래픽에 대한 정보는 파악하기가 힘들게 된다는 단점이 있다[8]. 그러므로 패킷을 마킹할 때 Win_length의 크기에 따른 편향된 정보를 토대로 마킹하게 된다. 이것은 TSW가 네트워크 안으로 유입되는 트래픽에 대한 정보를 정확하게 파악하지 못하여 공평한 대역폭 분배에 한계를 가질 수 있음을 의미한다.

3. 개선된 마킹 알고리즘 제안

기존의 TSW 마킹 알고리즘에서는 측정된 평균전송률이 Win_length의 값에 따라 서로 다르게 편향적으로 해석될 수 있다.

제안된 알고리즘의 구성 요소는 TSW와 마찬가지로 데이터의 전송률을 측정하는 전송률 측정기(rate estimator)와 패킷에 우선순위를 마킹하는 마커로 이루어져 있으나, (그림 4)에 나타난 것처럼 전송률 측정기에서 전송률을 측정하는 window를 1개 더 추가하여 하나는 짧은 시간동안 전송률의 변화추이를 파악하고, 다른 하나는 상대적으로 긴 시간동안의 전송률을 측정하여, 일정기간동안의 평균 전송률을 파악 할 수 있도록 하였다.

```

초기값 :

L_Win_length      = a constant ;
L_Avg_rate        = target rate, RT ;
S_Win_length      = a constant ;
S_Avg_rate        = target rate, RT × constant ;
T_front          = 0 ;

매패킷 도착시 :

L_Bytes_in_TSW   = L_Avg_rate × L_Win_length ;
L_New_bytes      = L_Bytes_in_TSW + pkt_size ;
L_Avg_rate       = L_New_bytes /
                  (Now - T_front + L_Win_length) ;

S_Bytes_in_TSW   = S_Avg_rate × S_Win_length ;
S_New_bytes      = S_Bytes_in_TSW + pkt_size ;
S_Avg_rate       = S_New_bytes /
                  (Now - T_front + S_Win_length) ;

T_front          = now ;
    
```

(그림 4) 개선된 전송률 측정 알고리즘

이러한 개념은 순간적인 트래픽의 변화에 대한 정보와 상당히 오랜 시간동안의 평균 전송률에 대한 정보를 같이 가질 수 있게 되어, 경계라우터에서 패킷을 마킹할 때보다 정확하게 판단을 할 수 있게 된다. 마커는 TSW에서와 마찬가지로 전송률 측정기에서 측정된 전송률을 기초로 해서 패킷을 마킹하게 되며, 제안된 알고리즘은 (그림 5)에 나타난 바와 같다.

제안된 알고리즘에서는 패킷을 마킹하기 위해 전송률 측정기에서 측정된 LR(long-term average rate)과 SR(short-term average rate)에 대하여 LIR(long-term committed information rate)과 SIR(short-term committed information rate) 값의 초과여부를 비교하게 된다. TCP 혼잡제어 방식을 기초로 해서 생각할 때, LR과 SR의 관계에 따라서 (그림 5)에 나와있는 것처럼 6가지로 트래픽의 유입상태를 유추할 수 있다.

①번의 경우 LR과 SR 두 가지 모두 LIR 보다 작은 경우로 패킷의 유입량이 계속 작은 상태로 TCP가 트래픽을 발생시키는 초기이거나, TCP가 발생시키는 트래픽의 양이 네트워크에서 충분히 소화할 수 있는 정도로 적은 것임을 알 수 있다.

```

Estimate sending rate LR and SR

if LR <= LIR
  calculate DR = SR * Pdrop
  if SR <= LIR ①
    mark packet as Green
  else if SR > LIR and SR < SIR ..... ②
    if DR > LIR ..... ④
      mark packet as Yellow
    else
      mark packet as Green
  else if SR > SIR ..... ③
    if DR > SIR ..... ⑥
      mark packet as Red
    else
      mark packet as Yellow
    
```

```

else
  calculate DR = LR * Pdrop
  if SR <= LIR ④
    if DR > LIR ..... ③
      mark packet as Yellow
    else
      mark packet as Green
  else if SR > LIR and SR < SIR ..... ⑤
    if DR > LIR ..... ④
      mark packet as Red
    else
      mark packet as Yellow
  else if SR > SIR ..... ⑥
    mark packet as Red

Pdrop : drop probability.
LR : long-term average sending rate.
SR : short-term average sending rate.
LIR : long-term committed information rate.
SIR : short-term committed information rate.
DR : calculated downgrade rate.
    
```

(그림 5) 개선된 마킹 알고리즘

②번의 경우는 LR이 LIR보다 작고, SR이 LIR보다는 큰 경우로, 오랜 시간 동안의 전송률은 낮고, 짧은 시간동안의 전송률은 어느 정도 높은 수준으로 패킷의 유입량이 점차 많아지고 있는 상태라는 것을 알 수 있다. ③번의 경우는 ②번과 비슷한 경우이지만 순간적인 유입량이 많은 상태로 트래픽이 급격하게 증가하는 상태임을 알 수 있다.

④번의 경우는 LR이 LIR보다 크고 SR이 LIR보다 작은 상태로, 트래픽의 유입량이 갑자기 줄어들었음을 알 수 있는데 이는 TCP가 congestion을 겪은 후의 회복상태에 있는 것으로 예상할 수 있다.

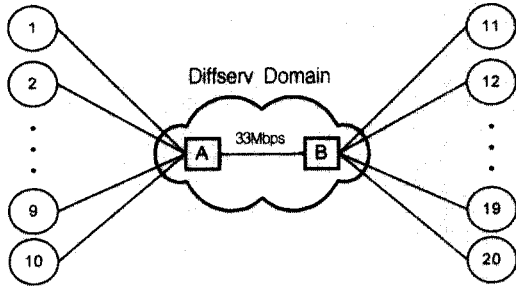
⑤번의 경우는 패킷의 유입량이 계속 많은 상태, 그리고 ⑥번의 경우는 폭주상태로 생각할 수가 있다. 따라서 앞에서 말한 6가지의 정보를 바탕으로 플로우의 상태를 보다 정확하게 판단하여 마킹에 이용할 수가 있다.

오랜 시간 동안의 전송률 LR이 LIR보다 작은 경우 트래픽의 유입량이 적은 상태로 ④와 ⑥에서처럼 패킷이 되도록 낮은 패킷 폐기 우선순위로 마킹되도록 유도하고, LR이 LIR 보다 큰 경우는 ③와 ④에서처럼 패킷이 되도록 높은 패기 우선순위로 마킹되도록 하여 라우터에서 혼잡상태가 발생할 경우 트래픽을 적극적으로 조절할 수 있다.

4. 시뮬레이션 및 결과분석

본 연구의 기본적인 시뮬레이션 환경은 (그림 6)에 나타난 토폴로지와 <표 1>에 나타난 파라미터를 기본으로 하였다. 송신측과 수신측에 각각 10개의 노드를 배치시켰으며, 각각의 노드는 상대방의 노드와 1대 1로 TCP Reno로 연결하였다. 송신측 노드와 노드 A, 수신측 노드와 노드 B는 공히 전송속도 10Mbps, 전송지연시간 20ms로 연결하였으며, 노드 A와 B는 전송속도 33Mbps, 전송지연시간 5ms

로 연결하여 공유하도록 하였다. 그리고 차별서비스 네트워크의 경계 라우터 큐 관리 알고리즘은 RIO를 사용하였다.



(그림 6) 모의실험을 위한 네트워크 구성

10개의 호스트는 5개의 소그룹으로 구성하여 같은 그룹에 속한 호스트는 동일한 예약 대역폭을 갖게 하였다. 예약 대역폭 30Mbps를 제외한 3Mbps는 모든 플로우들이 공유하도록 하였다. 이렇게 할 경우 플로우 i의 예약 전송률(대역폭)을 R_i , 백본링크의 용량을 C, 백본링크를 사용하는 플로우의 수를 n, 여분의 대역폭을 e라고 할 때 <표 2>에 제시된 목표 전송률은 다음과 같이 계산될 수 있다.

전체 예약 전송률 R_T 는 $\sum_{i=1}^n R_i$, 잉여 대역폭 e는 $C - \sum_{i=1}^n R_i$ 그리고 각 플로우별 목표 전송률 R_i 는 $R_i + \frac{e}{n}$ 이다[9].

시뮬레이션 수행에 사용된 패킷의 크기는 1000Byte로 하였으며, 경계 라우터의 RIO 파라미터는 <표 1>에 제시하였다.

TSW의 Win_length는 1sec로 설정하였으며, 제안한 알고리즘의 Win_length는 Short-term Win_length를 0.3sec, Long-term Win_length를 1.2sec로 설정하였다. 제안된 알고리즘의 SIR(short-term committed information rate)는 식 (1)과 같이 각 플로우의 예약전송률에 상수를 곱하여 설정하였으며, 본 논문에서는 일차적으로 상수 값을 2.0으로 설정하였다.

$$SIR = LIR(RT) * SIR\ factor \quad (1)$$

<표 1> RIO 파라미터 값

TSW, RIO(2 drop)			
IN		OUT	
Min _{th}	64	Min _{th}	32
Max _{th}	128	Max _{th}	64
P _{drop}	0.02	P _{drop}	0.2

TS2W3C, RIO(3 drop)					
Green		Yellow		Red	
Min _{th}	64	Min _{th}	48	Min _{th}	32
Max _{th}	128	Max _{th}	96	Max _{th}	64
P _{drop}	0.02	P _{drop}	0.13	P _{drop}	0.2

시뮬레이션은 30초동안 실행하였으며, 패킷의 처리율은 TCP가 안정상태에 도달한 시뮬레이션 시작 후 5초부터 25초

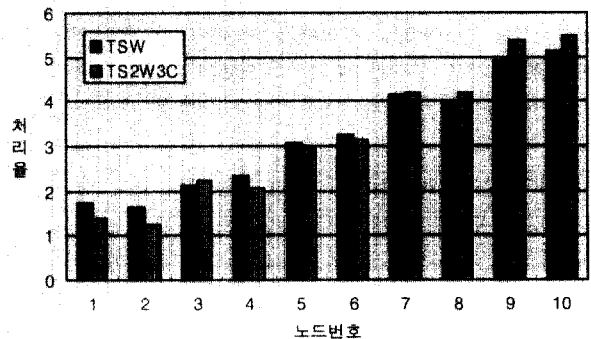
간 측정하였다. 시뮬레이션의 결과를 <표 2>에 정리하였다.

ns-2를 이용한 시뮬레이션을 통해 기존의 TSW 알고리즘과 본 논문에서 제안한 TS2W3C 알고리즘의 성능을 비교 분석한 결과, 목표 전송률에 대한 도달 비율의 표준편차가 본 논문에서 제안한 방법이 0.058, 그리고 기존의 TSW 알고리즘이 0.148로 본 논문에서 목표로 했던 전송률 공평성 향상측면에서 개선된 성능을 보여 주었다.

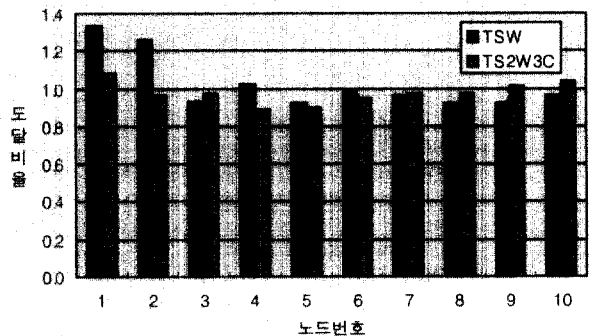
<표 2> 성취된 처리율(단위 : Mbps)

flow	예약 전송률	목표 전송률	처리율(Throughput)	
			TS2W3C	TSW
1	1	1.3	1.40000	1.73088
2	1	1.3	1.25248	1.63552
3	2	2.3	2.24000	2.13152
4	2	2.3	2.04960	2.34496
5	3	3.3	2.95072	3.05472
6	3	3.3	3.13504	3.24096
7	4	4.3	4.19520	4.15968
8	4	4.3	4.19616	3.94848
9	5	5.3	5.37536	4.90848
10	5	5.3	5.49248	5.11712
합계	30	33	32.25824	32.27232

<표 2>의 결과를 분석해보면 TSW 알고리즘의 경우 예약 전송률에 따른 처리율의 편향현상이 크게 나타난 반면, 새로 제안한 알고리즘의 경우 예약전송률에 따른 처리율의 편향현상이 줄어들어 비교적 공평성이 향상되었음을 알 수 있다.



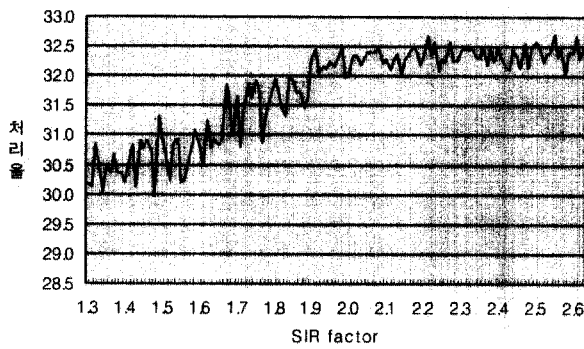
(그림 7) TSW와 TS2W3C 알고리즘의 플로우별 처리율 비교



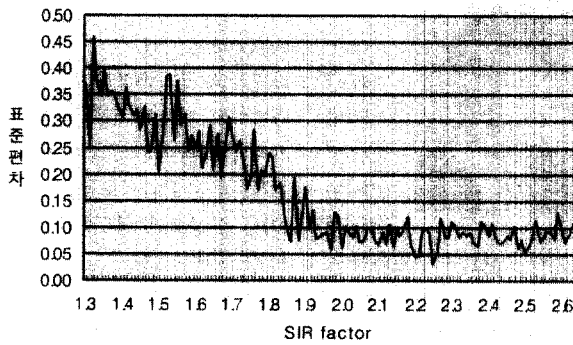
(그림 8) TS2W3C와 TSW의 목표전송률에 대한 도달 비율 비교

$$\text{도달비율} = \text{처리율} / \text{목표전송률} \quad (2)$$

(그림 7)에는 플로우별 처리율의 비교 그래프를, 그리고 (그림 8)에는 식 (2)에 나타난 각 플로우별 목표 전송률에 대한 처리율을 도달비율로 나타내었다. (그림 9), (그림 10)은 제안한 알고리즘에서 최적의 SIR factor를 찾기 위해 SIR factor를 1.33부터 2.66까지 변화시키면서 처리율 및 목표 전송률에 대한 플로우별 도달비율의 표준편차를 나타낸 것으로 SIR factor의 값이 1.9이상에서 처리율 및 플로우별 도달비율의 표준편차가 비교적 안정적인 상태로 접어드는 현상을 발견할 수 있었다.



(그림 9) SIR factor에 따른 처리율



(그림 10) SIR factor에 따른 목표전송률에 대한 노드별 도달비율의 표준편차

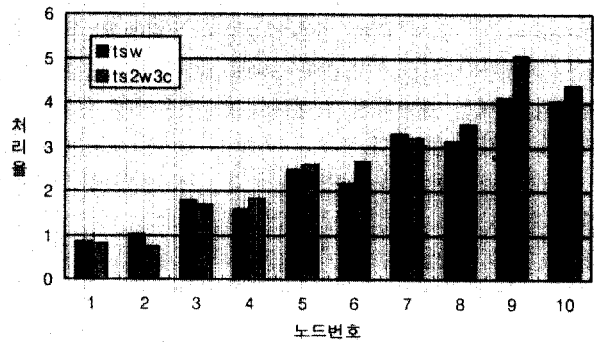
5. 무응답 플로우의 영향

무응답 플로우는 TCP의 congestion 알고리즘을 따르지 않기 때문에 네트워크에서 패킷이 폐기되더라도 자신의 전송속도를 낮추지 않는다. 따라서 무응답 플로우는 네트워크에 congestion 상황을 발생시키고 공정성을 저하시키는 주요 원인으로 작용한다[6].

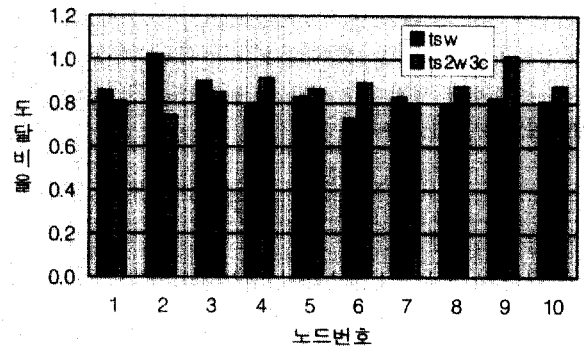
무응답 플로우의 영향을 알아보기 위하여 4절에서 사용하였던 기본적인 시뮬레이션 환경과 동일한 상태에서 트래픽을 10Mbps로 발생시키는 UDP 소스를 포함하였다. UDP 소스의 예약전송률은 노드별 예약전송률의 평균값인 3Mbps로 설정하여 시뮬레이션 하였다.

<표 3> UDP 소스를 포함한 경우의 플로우별 처리율(단위 : Mbps)

flow	예약 전송률	목표 전송률	처리율(Throughput)	
			TS2W3C	TSW
1	1	1	0.80480	0.85536
2	1	1	0.74336	1.02496
3	2	2	1.69760	1.79616
4	2	2	1.82304	1.56704
5	3	3	2.59520	2.48416
6	3	3	2.67744	2.18016
7	4	4	3.19168	3.30336
8	4	4	3.52288	3.13248
9	5	5	5.05696	4.09984
10	5	5	4.39136	4.03040
11	3	3	6.49408	8.48320
합계	33	33	32.9984	32.95712



(그림 11) UDP 소스를 포함한 경우의 플로우별 처리율 비교



(그림 12) UDP 소스를 포함한 경우의 플로우별 도달비율 비교

<표 3>에 UDP 소스를 포함한 시뮬레이션 결과를 플로우별로 나타내었으며, (그림 11)에는 플로우별 처리율의 비교 그래프를, 그리고 (그림 12)에는 플로우별 목표전송률에 대한 도달비율을 나타내었다.

UDP 소스를 포함한 시뮬레이션에서 기존의 TSW 알고리즘과 본 논문에서 제안한 TS2W3C 알고리즘의 성능을 비교 분석한 결과, UDP 소스를 제외한 노드의 전체 전송률이 26.5Mbps와 24.4Mbps로 본 논문에서 제안한 방법의 전송률이 더 높게 나왔으며, 목표 전송률에 대한 도달 비율

의 표준편차 또한 본 논문에서 제안한 방법이 0.072, 그리고 기존의 TSW 알고리즘이 0.080으로 본 논문에서 제안한 방법이 전송률 및 노드별 전송률의 공평성을 향상시킨 것으로 나타났다.

6. 결 론

본 논문에서는 차별화 서비스에서 보장형 서비스의 전송률 공평성에 대한 문제를 다루었다. 기존의 TSW 알고리즘의 문제점을 분석하고, 플로우들의 예약대역폭에 따른 대역폭 분배의 편향현상을 완화시킬 수 있는 개선된 알고리즘을 제안하였다. 기존의 TSW 알고리즘은 전송률을 측정하는 윈도우가 하나뿐이기 때문에 윈도우의 크기에 따라 전송률에 대한 정보를 편향적으로 해석될 수 있기 때문에 패킷 마킹시에 현재의 상태를 정확하게 반영할 수가 없다.

제안된 알고리즘은 TSW 알고리즘에 윈도우를 하나 더 추가하여 전송률에 대한 평가를 2가지 기준으로 평가함으로써 패킷을 마킹할때 기존의 TSW 알고리즘보다 현재 트래픽의 상태를 정확하게 반영하여 좀더 유연성 있고 적극적으로 마킹할 수 있다.

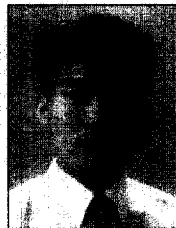
ns-2를 이용한 시뮬레이션을 통해 기존의 TSW 알고리즘과 본 논문에서 제안한 알고리즘의 성능을 비교 분석한 결과, 목표 전송률에 대한 도달 비율의 표준편차가 본 논문에서 제안한 방법이 0.058, 그리고 기존의 TSW 알고리즘이 0.148로 본 논문에서 목표로 했던 전송률 공평성 향상 측면에서 개선된 성능을 보여 주었다.

향후 연구과제로는 네트워크 부하 정도에 따른 SIR factor의 결정에 관한 연구가 필요하며, 윈도우의 크기에 따른 성능의 차이를 분석하여 적절한 윈도우크기를 찾아내는 것이다. 또한 다양한 네트워크 구조 및 트래픽 소스에 대한 성능평가가 필요하다.

참 고 문 헌

[1] S. Blake, et al., An Architecture for Differentiated Services, Internet RFC 2475, Dec., 1998.
 [2] N. Seddigh, et al., Bandwidth Assurance Issues for Tcp flows in a Differentiated Services Network, Globecom '99, Dec., 1999.
 [3] D. Clark, W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service," ACM Transactions on Networking, Aug., 1998.
 [4] Network simulator(Ns), University of California at Berkeley, CA, 1997. Available via <http://www-nrg.ee.lbl.gov/ns/>.

[5] J. Heinanen, R. Guerin, "A Two Rate Three Color Marker," Internet RFC 2698, September, 1999.
 [6] S. Floyd, K. Fall, "Promoting the Use of End-to-End Congestion control in the Internet," IEEE/ACM Transactions on Networking, May, 1999.
 [7] I. Yeom and N. Reddy, Impact of marking strategy on aggregated flows in a DiffServ network, IEEE ICMCS '99, Dec., 1999.
 [8] W. Lin, R. Zheng and J. Hou, How to Make Assured Services More Assured, ICNP '99, Oct., 1999.
 [9] I. Yeom, J. Hou, Realizing Throughput Guarantees in a Differentiated Services Network, ICMCS, Florence, Italy, June, 1999.



조 병 규

e-mail : bkcho@sunchon.ac.kr
 1999년 순천대학교 컴퓨터교육과(이학사)
 2002년 순천대학교 대학원(이학석사)
 1999년~2001년 순천대학교 컴퓨터 교육과 조교
 2002년~현재 순천대학교 사범대학 조교
 관심분야 : 인터넷 QoS, 네트워크 보안, 전자상거래



이 성 근

e-mail : sklee@sunchon.ac.kr
 1985년 고려대학교 전자공학과(공학사)
 1987년 고려대학교 대학원(공학석사)
 1995년 고려대학교 대학원(공학박사)
 1987년~1992년 삼성전자 정보통신 연구소
 1996년~1997년 삼성전자 네트워크개발팀
 1997년~2002년 순천대학교 컴퓨터교육과 조교수
 2002년~현재 순천대학교 정보통신공학부 조교수
 관심분야 : 인터넷 QoS, 분산 멀티미디어, 광대역 통신 프로토콜



강 의 성

e-mail : e.kang@ieee.org
 1991년 고려대학교 전자전공학과 학사
 1995년 고려대학교 전자공학과 석사
 1999년 고려대학교 전자공학과 박사
 1999년~2000년 고려대학교 전자공학과 Post-Doc.
 2000년~2001년 (주)비트닉스 연구소장
 2001년~현재 순천대학교 컴퓨터교육과 전임강사
 관심분야 : 신호 및 영상처리, HCI, 멀티미디어 응용