

공평성 향상을 위한 TSW Tagging 알고리즘

김성용^o 김영주 이기현
명지대학교 컴퓨터공학과
{ksy75^o, youngju, khlee}@mju.ac.kr

TSW Tagging Algorithm for Improving Fairness

SungYong Kim^o YoungJoo Kim, keeHyun Lee
Dept. of Computer Engineering, Myeong-Ji University

요 약

인터넷의 발전으로 인하여 여러 가지 서비스가 가능하게 되었고, 이러한 서비스들은 서로 다른 서비스 품질(QoS)이 요구된다. 따라서 이러한 QoS를 제공하기 위해 차별화 서비스(Differentiated Service)의 연구가 진행되고 있다. 이 논문에서는 차별화 서비스중 보장형 서비스에서 공평성에 관한 문제로 지적되고 있는 bandwidth skew problem의 문제를 최소화하기 위한 기법을 제시한다. 이 논문에서는 bandwidth skew problem의 현상을 확인하고 제안한 기법이 기존의 기법보다 공평성 측면에서 개선된 성능을 제공함을 보였다.

1. 서 론

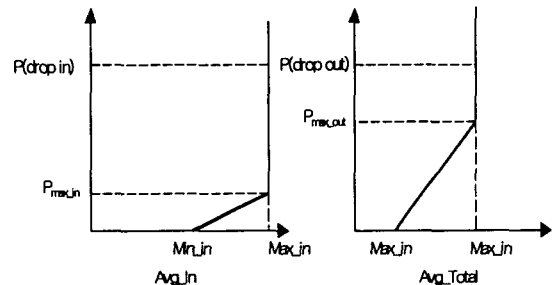
인터넷의 발전으로 인하여 여러 가지 서비스가 가능하게 되었다. 이러한 서비스들은 서로 다른 전송 특성과 서로 다른 서비스품질(QoS)을 요구한다. 서비스의 품질을 보장하는 방법에는 크게 통합 서비스(Integrated Service)와 차별화 서비스(Differentiated Service)로 나눌 수 있다. 통합 서비스는 RSVP(Resource Reservation Protocol)를 통해 자원을 예약하고 서비스의 품질을 보장한다. 그러나 RSVP는 라우터가 각 플로우의 정보를 관리해야 하므로 대규모 네트워크에서는 바람직하지 못하다. 차별화 서비스는 경계라우터(Edge Router)에서 패킷분류(classifying), 미터링(metering), 셰이핑(shaping) 등과 같이 복잡한 기능을 수행하고, 중심라우터(Core Router)에서 간단한 포워딩 기능을 수행함으로써 이러한 문제점을 해결할 수 있다. 차별 서비스 네트워크는 세 가지의 서비스 클래스 즉 프리미엄 서비스(Premium Service), 보장형 서비스(Assured Service), 최선형 서비스(Best-Effort Service)로 나누어진다. 이 논문에서 다루어질 보장형 서비스는 네트워크에서 혼잡상황이 발생하여 처리율(throughput)이 감소하더라도 약속된 처리율을 보장해 주는 방식이다[1].

2. 배경지식

2.1 코어 라우터에서의 차별화된 드롭 알고리즘

차별화 서비스의 코어 라우터에서는 RIO(RED with In and out)를 사용하여 차별화된 패킷 마킹을 할 수 있다[2]. RIO는 IN 패킷과 OUT 패킷 각각에 Min_{th} , Max_{th} , P_{max} 세 개의 파라미터를 사용하여 차별화된 드롭 확률을

제공한다. (그림 1)은 RIO알고리즘을 나타낸 것이다. 만약 IN 패킷이 큐에 진입 하였을 때에는 avg_{in} 을 계산하여 평균 큐의 길이를 계산하고, OUT 패킷이 큐에 진입 하였을 때에는 avg_{total} 을 계산하여 평균 큐의 길이를 계산한다. 이렇게 계산된 평균 큐의 길이와 미리 정의된 P_{max} 를 이용하여 패킷의 드롭 확률을 계산한다. 큐에 입력되는 패킷의 증가로 인해 평균 큐의 길이가 증가한다면 Min_{out} 이 Min_{in} 보다 작기 때문에 OUT 패킷이 먼저 드롭이 일어나고 또한 $P_{max_{out}}$ 이 $P_{max_{in}}$ 보다 크기 때문에 OUT 패킷이 더 높은 확률로 드롭 된다.



(그림 1) RIO algorithm

2.2 경계 라우터에서의 차별화된 태깅 알고리즘

경계 라우터에서는 TSW(Time-Sliding Window)를 사용하여 평균 전송률을 측정하고 약속된 전송률 CIR(Committed Information Rate)을 넘었을 경우 $(avgrate - CIR)/CIR$ 의 확률로 OUT패킷을 태깅한다. (그림 2)는 TSW 알고리즘을 나타낸 것이다[2].

```

rate estimator algorithm
Initially:
Win_length = a constant;
Avg_rate = connection's target rate, CIR;
T_front = 0;

Upon each packet arrival, TSW update it state variables as follows:

Bytes_in_TSW = Avg_rate * Win_length;
New_bytes = Bytes_in_TSW + pkt_size;
Avg_rate = New_bytes / (now - T_front + Win_length);
T_front = now;
tagging algorithm
if avg_rate < CIR
    All packets are tagged as In
else
    tag packets as out with p = (avg_rate - CIR) / CIR
    
```

(그림 2) TSW tagging algorithm

저 connection 수를 고려하여 connection 당 대역폭 (Bandwidth/Connection)을 계산하고 CIR이 B/C 이상일 경우 CIR의 120%의 비율로 재설정해 주고 CIR이 B/C 미만하일 경우 CIR의 70%의 비율로 재설정하였다. 따라서 CIR이 낮은 connection일 경우 더 많은 OUT패킷을 생성하게 된다. (그림 3)은 이 논문이 제안한 TSW tagging 방식을 나타낸 것이다

```

tagging algorithm
if (CIR ≥ bandwidth / connection수)
    if(avg_rate ≥ CIR * 1.2)
        tag packets as out with P = (avg_rate - CIR) / CIR
    else
        .....
else
    if(avg_rate ≥ CIR * 0.7)
        tag packets as out with P = (avg_rate - CIR) / CIR
    
```

(그림 3) 제안한 TSW tagging algorithm

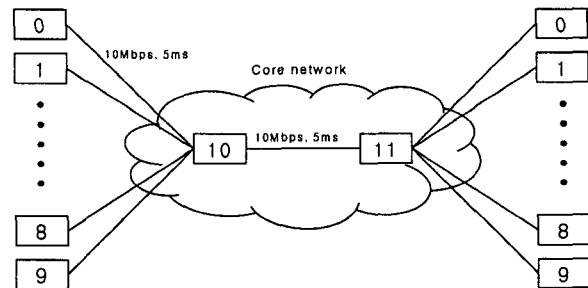
3. TSW 알고리즘에서의 공정성 문제

TCP의 혼잡 윈도우(Congestion Window)는 slow start와 fast recovery를 반복적으로 수행한다. slow start는 retransmission timeout이 일어났을 때 혼잡 윈도우를 1로 감소하고, 미리 정의된 ssthresh(Slow-Start Threshold) 까지 혼잡 윈도우를 지수적으로 증가시킨다. 그 이후에는 혼잡 윈도우를 선형적으로 증가시킨다. fast recovery는 3개의 중복 ACK(Duplicated Acknowledgements)을 받았을 경우 혼잡 윈도우를 반으로 감소시키고 혼잡 윈도우 크기를 1 RTT마다 1씩 증가 시킨다[3]. 이러한 TCP의 동작 때문에 혼잡윈도우의 모양이 톱니 모양으로 형성된다. 전송률이 작은 connection은 혼잡 윈도우의 감소하는 폭도 작아지게 된다. 따라서 CIR이 작은 플로우는 CIR이 큰 플로우보다 작은 RTT시간동안 CIR에 도달하게 된다. 예를 들어 윈도우 크기가 1KB이고 CIR이 각각 16KB 와 32KB를 가지는 두개의 플로우가 혼잡상황이 발생하여 fast recovery 상태에 들어갔을 때, 혼잡 윈도우의 크기는 각각 8KB, 16KB가 된다. 이때 원래의 CIR으로 도착하기 위해서는 각각 8 RTT와 16 RTT의 시간이 걸리게 된다. 따라서 CIR이 작은 플로는 더 빨리 CIR에 도달할 뿐만 아니라 잉여 대역폭까지 사용하여 CIR 보다 높게 전송 할 수 있게 된다. 이러한 문제를 bandwidth skew problem이라 한다[1].

4. 제안된 차별화 태깅 기법

이 논문에서는 위에 언급한 bandwidth skew problem을 해결하기 위해서 차별화된 태깅 기법을 제안한다. 먼

5. 시뮬레이션 및 결과 분석



(그림 4) Simulation Topology

(그림 4)는 이 논문의 시뮬레이션 환경의 토폴로지를 나타낸다. 송신측과 수신측은 각각 10개 노드로 구성되며, 각 노드는 1대 1로 연결되어 있다. 송신측과 수신측의 대역폭은 모두 10Mbps 속도와 5ms의 지연시간으로 같게 해 줌으로써, 예약 전송률 외의 변수를 통제하였다. 각 노드는 0Kbps, 100Kbps, 200Kbps, 300Kbps, 400Kbps, 1000Kbps, 1200Kbps, 1400Kbps, 1600Kbps, 2000Kbps로 예약 전송률을 설정하였다. 총 예약 전송률은 8200Kbps 이며 모든 노드가 공정하게 잉여 대역폭을 사용할 수 있다고 가정하면 각 노드는 최대 180Kbps정도의 속도를 더 가질 수 있다. RIO 파라미터는 $Min_{th_in}=40$, $Max_{th_in}=70$, $P_{max_in}=0.02$, $Min_{th_out}=10$, $Max_{th_out}=30$, $P_{max_out}=0.2$ 로 설정하였고 시뮬레이션 시간은 총 300초로 하였다. <표 1>은 기존 방식의 시뮬레이션 결과를 나타낸 것이다. CIR이 대역폭의 1/10 이하인 0번에서 4번 노드까지는 각각의 예상속도(CIR+잉여대역폭)보다도 높은 속도로 전송하는 것을 볼 수 있다. 반면에

7번에서 9번 노드는 CIR보다도 낮은 전송률을 보인다. 이것은 CIR이 낮은 노드들 대부분이 CIR이 높은 노드들의 잉여대역폭까지 사용했다는 것을 보여준다.

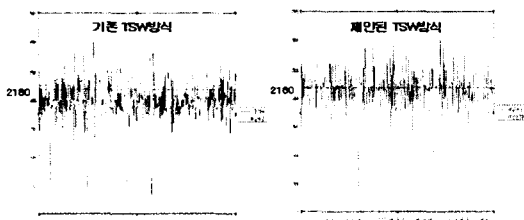
<표 1> 기존 TSW Tagging 방식

node No	CIR(Kbps)	예상속도	실제속도	증가율(%)
0	0	180	353.813	96.56
1	100	280	423.680	51.31
2	200	380	528.213	39.00
3	300	480	602.320	25.48
4	400	580	721.440	-24.39
5	1000	1180	1169.653	-0.88
6	1200	1380	1324.027	-4.06
7	1400	1580	992.507	-37.18
8	1600	1780	1171.253	-34.20
9	2000	2180	2002.880	-8.12

<표 2> 제안한 TSW Tagging 방식

node No	CIR(Kbps)	예상속도	실제속도	증가율(%)
0	0	180	317.840	76.58
1	100	280	394.587	40.92
2	200	380	490.293	29.02
3	300	480	569.760	18.70
4	400	580	655.787	13.07
5	1000	1180	1156.533	-1.99
6	1200	1380	1381.680	0.12
7	1400	1580	1070.533	-32.24
8	1600	1780	1227.253	-31.05
9	2000	2180	2134.933	-2.07

<표 2>는 제안한 방식으로 실험한 결과이다. <표 1>과 <표 2>를 비교해 보면 노드4의 경우 속도가 65.653Kbps 감소한 반면, 노드9의 경우는 132.053Kbps가 증가한 것을 볼 수 있다. 전체적으로 CIR이 1000Kbps 미만인 connection은 201.2Kbps 속도가 감소하였고, CIR이 1000Kbps 이상인 connection은 310.613Kbps 속도가 증가하였다. (그림 5)는 노드9의 초당 전송률의 변화를 나타낸 것이다.



(그림 5) 노드9의 초당 전송률의 변화

(그림 5)에서는 기존의 TSW방식보다 제안한 TSW방식이 예상속도(2180Kbps)에 더 근접함을 나타낸다. 이것은 예상속도를 기준으로 아래쪽으로 많이 형성되어 있는 기존의 TSW방식보다 제안한 TSW방식이 더 나은 보장된 서비스를 제공할 수 있음을 보여준다.

6. 결 론

이 논문에서는 NS-2 시뮬레이터를 사용하여 차별화 네트워크에서의 bandwidth skew problem을 보였고, 이것을 해결하기 위하여 차별화된 태깅 방식을 제안하였다. 제안한 방식으로 실험한 결과 CIR이 1000Kbps 이상인 connection은 속도가 4.66% 증가하였으며, CIR이 1000Kbps 미만인 connection은 속도가 7.65% 감소하여, 제한한 방식이 기존의 방식보다 공정성 측면에서 개선된 성능을 보였다. 그러나 여전히 CIR이 큰 connection이 예약 전송률에 도달하지 못하는 문제점이 남아있다. 또한 고정된 값이 아닌 네트워크상황에 따라서 최적값을 찾는 문제도 계속 연구 할만한 가치가 있다.

참고문헌

- [1] 조병규, 이성근, 강의성 “차별서비스 네트워크에서 보장형 서비스의 공정성 향상을 위한 이중 윈도우 기반 마킹 알고리즘”, 정보처리학회논문지 C 제9-C권 제5호, 2002.10
- [2] David D. Clark, Wenjia Fang, “Explicit Allocation of Best-Effort Packet Delivery Service”, IEEE/ACM Transaction on Networking, VOL.6, No.4 August 1988
- [3] S.Floyd, ACIR, “Congestion Control Principles”, Internet RFC 2914, September 2000.
- [4] Sally Floyd, Van Jacobson, “Random Early Detection Gateways for Congestion Avoidance”, IEEE/ACM Transactions on Networking, VOL.3 August 1993
- [5] 남동호, 최영수, 김병철, 조유제, “인터넷 차별화 서비스를 위한 라우터의 공정성 향상 알고리즘”, 정보과학회논문지, 정보통신 제29권 제4호, 2002.8
- [6] 최상기, 정광수, “가상 트래픽 프로파일을 이용한 대역 분배 메카니즘”, 정보과학회논문지, 정보통신 제27권 제3호 2000.9
- [7] Wu-Chang Feng, Dilip D. Kandlur, Debanjan Saha, Kang G. Shin, “A-Self-Configuring RED Gateway”, Proceedings of INFOCOM 99