

차별 서비스 네트워크에서 대역폭 스큐 문제를 완화하는 큐 관리 기법 연구

명성원^{*} · 황구연^{**} · 이성근^{*}

^{*}순천대학교, ^{**}순천제일대학

A Study on a Queue Management for Alleviating the Bandwidth Skew Problem in Differentiated Service Network

Sung-Won Myung^{*} · Goo-youn Hwang^{**} · Sung-keun Lee^{*}

^{*}Dept. of Computer Education, Suncheon National University

^{**}Dept. of Internet Information, Suncheon First College

E-mail : floydwon@hanmail.net

요 약

차별서비스 네트워크에서 큐 관리 알고리즘으로 RIO가 많이 적용되고 있다. RIO는 RED로부터 확장된 큐 관리 기법으로, IN 패킷과 OUT 패킷을 구분하여 차별적으로 처리한다. 보장형 서비스를 제공하기 위해 RIO를 큐 관리 메커니즘으로 적용했을 때 대역폭 스큐 문제가 발생된다. 본 논문에서는 대역폭 스큐 문제를 완화하기 위한 큐 관리 기법을 제안하고 시뮬레이션 방법을 통해 성능 분석을 수행한다.

ABSTRACT

In Diffserv Network, RIO is the most effective active queue management algorithm. RIO, the enhanced queue management algorithm from RED, divides IN and OUT packets and manages them differently. one of the main problems for the implementation of RIO is the Bandwidth Skew problem for providing the Assured Service. In this paper, we will show the bandwidth skew problem in RIO and provide an enhanced RIO mechanism that will alleviate the bandwidth skew problem.

키워드

차별 서비스, RED(Random early Drop), RIO(RED with IN and OUT)

1. 서 론

차별서비스는 QoS(Quality of Service)를 보장하기 위해 IETF에서 제안한 네트워크 모델이다. 차별서비스 네트워크는 내부 라우터에서 제공하게 되는 PHB(Per Hop Behavior)에 따라서 네트워크의 QoS를 보장한다.[4] PHB는 패킷을 그룹화시켜 그룹에 따른 차별적인 처리를 하는 것이다. 차별 서비스 네트워크의 구성은 그림 1과 같다.

PHB는 내부 네트워크에 위치한 기능으로서 서비스 그룹에 따라 패킷을 처리하는 행위로서 일반적으로 큐 관리 메커니즘을 의미한다[4].

서비스 제공자와 사용자 사이에는 서비스 등급에 대한 계약을 하게 되는데 이것을 SLA(Service Level Agreement)라고 한다.

IETF(Internet Engineering Task Force)에서는 PHB를 프리미엄 서비스, 보장형 서비스, 그리고 최선형 서비스로 분류하고 있다. 프리미엄 서비스는 네트워크가 혼잡상태가 발생해도 서비스 공급자와 사용자 사이에 계약된 전송률을 확실하게 보장해 주는 서비스이고, 보장형 서비스는 계약된 전송률을 보장하면서 동시에 여분의 대역폭을 공정하게 분배하는 서비스이다. 최선형 서비스는 현재 인터넷 서비스와 같이 QoS를 보장하지 않는다.

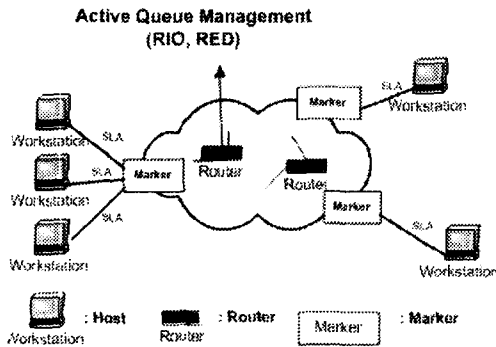


그림 1. 차별 서비스 네트워크 구성도

보장형 서비스를 구현하기 위해 현재 가장 많이 적용되는 큐 관리 알고리즘인 RIO[5]를 분석한 Ibanez와 Nichols[1]는 예약 전송률의 값이 큰 흐름들이, 예약 전송률의 값이 낮은 흐름들에 비해 전송률이 상대적으로 낮다는 것을 발견했다.

박원형의 논문[2]에서도 예약 전송률이 낮은 흐름이 목표 전송률에 더 근접하게 달성한다고 지적하였고, 이러한 현상을 대역폭 스큐 문제라고 정의하였다.

본 논문에서는 이러한 대역폭 스큐 문제를 완화하기 위한 확장된 RIO를 제시한다. 본 논문의 구성은 다음과 같다. 2장에서는 RIO에서 나타난 대역폭 스큐 문제를 언급하고 3장에서는 확장된 RIO 알고리즘을 제안한다. 4장에서 시뮬레이션으로 기존의 RIO와 확장된 RIO의 성능을 분석하고 마지막 5장에서 결론을 맺는다.

II. RIO상의 대역폭 스큐 문제

Ibanez와 Nichols[1]에서 처음 제기되고, 박원형 [2]에서 구체적으로 설명된 이 대역폭 스큐 문제는 차별서비스의 내부 라우터에 RIO를 구현하기 위해 해결해야할 중요한 문제로 부각되었다.

2.1 RIO

RED[6]를 기반으로 한 큐 관리 기법으로서 현재 가장 각광을 받고 있는 큐 관리 기법이다. RIO는 IN 패킷과 OUT 패킷 각각에 RED 알고리즘을 적용하여 IN 패킷과 OUT 패킷을 차등적으로 처리한다. 큐에는 IN 패킷과 OUT 패킷에 대해 각각의 MIN_{th} 와 MAX_{th} 를 두어 IN 패킷의 평균율과 OUT 패킷의 평균 개수에 따라 폐기 확률이 가중치를 주어 임의적으로 폐기를 발생하게 하는 알고리즘이다.[5] 입력되는 패킷이 IN 패킷일 때, 패킷의 평균 개수에 따라 패킷의 개수가 $MIN_{in, th}$ 이하이면, 패킷을 통과시키고, $MAX_{in, th}$ 와 $MIN_{in, th}$ 사이면 임의적으로 폐기 확률을 높여서 임의적인 패킷 폐기를 수행한다. 망 혼잡이 발생하는 $MAX_{in, th}$ 를 넘어서면 모든 패킷을 무조건

폐기하여 망 혼잡 발생시에도 예약 전송률을 보장하기 위한 알고리즘이다. 그림 2는 RIO의 알고리즘을 나타냈다.

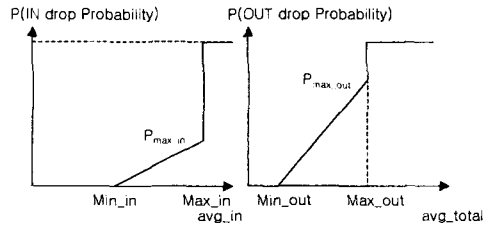


그림 2. RIO 알고리즘

2.2 RIO 상의 대역폭 스큐 문제 분석

Ibanez와 Nichols[1]의 논문에서는 RIO를 적용하여 네트워크에서 영향을 미치는 RTT(round trip time)와 계약된 전송률(목표 전송률)의 값, 그리고 패킷 크기에 따른 분석을 했다. 여기서 목표 전송률이 작은 값을 가진 흐름들이 높은 목표 전송률을 가진 흐름들에 대해 상대적으로 높은 전송률을 나타내었다는 것을 지적했고, 박원형[2]은 이 문제를 대역폭 스큐 문제라고 정의 했다.

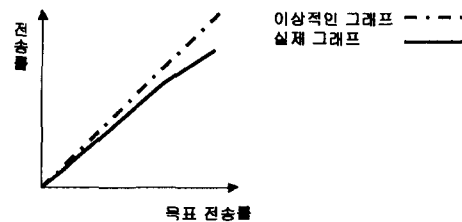


그림 3. 이상적인 전송률 vs 실제전송률

그림 3은 이상적인 전송률과 실제 결과 전송률에 대해 도식화 한 것이다.

대역폭 스큐 문제에 대해 시뮬레이션 방법을 통해 분석하고자 한다. 시뮬레이션 환경 및 네트워크 구조는 그림 4와 같다

백본의 대역폭은 10Mbps이고, 흐름은 20개이며, 이중 10개가 최선형 서비스이고 나머지 10개는 보장형 서비스이다. 경계 라우터의 마킹 알고리즘은 TSW(Time Sliding window)를 적용하였고, Round Trip Time은 15ms를 주었다. 시뮬레이션은 90초를 시행하고, 이중 TCP가 안정상태에 도달하는 5초 이후부터 85초간 TCP의 sink에 수신된 패킷 전송률을 측정하였다.

TCP는 현재 가장 많이 적용되는 Reno를 이용하였고 ftp 트래픽을 사용하였다. RIO에서 사용된 파라미터는 표 1에 나타내었다.

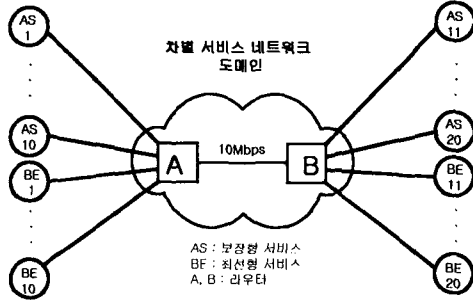


그림 4. 시뮬레이션 네트워크 구조

표 1. RIO 파라미터 값

| 파라미터 | 값 |
|----------------|------|
| $MIN_{out,th}$ | 10 |
| $MAX_{out,th}$ | 30 |
| P_{maxout} | 0.2 |
| $MIN_{in,th}$ | 40 |
| $MAX_{in,th}$ | 70 |
| P_{maxin} | 0.02 |

각각의 보장형 서비스 흐름들에게 제공하는 목표 전송률과 측정된 전송률은 표 2와 같다

표 2. AS의 Target Rate 와 전송률

| Assure Service | | | Best Effort | |
|----------------|-------------|----------|-------------|----------|
| flow | Target Rate | 전송률 | flow | 전송률 |
| 1 | 0.1M | 0.269161 | 11 | 0.148653 |
| 2 | 0.1M | 0.251354 | 12 | 0.160079 |
| 3 | 0.2M | 0.348145 | 13 | 0.151025 |
| 4 | 0.2M | 0.363411 | 14 | 0.136964 |
| 5 | 0.5M | 0.643317 | 15 | 0.16695 |
| 6 | 0.5M | 0.627072 | 16 | 0.149463 |
| 7 | 1M | 1.062987 | 17 | 0.151345 |
| 8 | 1M | 1.032455 | 18 | 0.156992 |
| 9 | 2M | 1.812484 | 19 | 0.161434 |
| 10 | 2M | 1.835279 | 20 | 0.14839 |

표 2를 분석한 결과, 목표 전송률이 높을수록 성취된 전송률의 값은 상대적으로 낮은 전송률을 보였다. 이 현상을 분석해 본 결과, 일단 한 패킷이 폐기 될 때, 작은 값의 목표 전송률을 가진 흐름들이 큰 값의 목표 전송률을 가진 흐름들에 비해 자신의 목표 전송률을 빨리 회복하여 백본에 자신의 OUT 패킷을 보낼 수 있는 시간이, 높은 목표 전송률을 가진 흐름에 대해 상대적으로 많았기 때문에 자신의 목표 전송률보다 높은 전송률을 획득할 수 있었다. 반면, 큰 목표 전송률을 가진 흐름들은 낮은 목표 전송률을 가진 흐름들에 비해, 일단 한 패킷의 폐기가 발생하면, 자신의 목표 전송률에 대한 회복에 걸리는 시간이 많이 필요하므로 상대적으로 전송률이 낮음을 알 수 있다. 이러한 결과는 [1]과 [2]에서 나타난 결

과와 같았다.

III. 개선된 RIO 알고리즘 제안

목표 전송률이 높은 흐름을 구분하기 위해 [2]에서는 전체 큐 길이에서 IN 패킷이 점유한 비율 값을 측정하였다. 이를 파라미터 C로 규정하고, 식 (1)과 같이 정의하였다.

$$C = \text{AVG}_{in} / \text{AVG}_{Total} \quad (1)$$

박원형은 한 패킷이 폐기 될 때, 높은 목표 전송률을 가진 흐름의 C값이, 낮은 목표 전송률을 가진 흐름의 C값에 비해 더 높은 값을 지녔다[2].

본 논문에서는 C값을 이용하여 OUT 패킷의 폐기 확률 P_{out} 을 제어하는, 대역폭 스쿠 문제의 영향을 완화시키기 위한 알고리즘을 제안한다. 그림 5는 대역폭 스쿠 문제를 완화하기 위한 제안 알고리즘이다.

```

for each arriving packet P
    calculate avgn
    calculate avgtotal
    if P is marked as IN
        the same RIO algorithm for IN
    else if P is marked as OUT
        if minth_out < avgtotal < maxth_out
            calculate Pout
            calculate C = avgn / avgtotal
            Pold = Pout
            if (C > k)
                Pnew = ((0.3 * Pold) * (C - k)) / (k - 1) + Pold;
            else
                Pnew = ((1.3 * Pold) - ((0.3 / k) * Pold * C));
            drop P with Probability Pnew
        else if avgtotal > maxth_out
            drop P
    
```

그림 5. 확장 알고리즘

기존의 RIO에서 IN 패킷 처리부분은 그대로 사용하고, OUT 패킷 처리시에는 C를 구하여 상수값 k에 따라서 P_{out} 의 값을 증감시킨다. 상수값 k는 이상적인 C값을 말한다. C값이 높을수록 P_{out} 값을 감소시켜 OUT패킷이 큐에 위치할 확률을 조절하고, C값이 낮을수록 P_{out} 값을 높여 OUT패킷의 처리율을 낮춘다. 따라서 상대적으로 IN 패킷의 처리율을 높여 흐름별로 자신의 목표 전송률을 달성하도록 한다.

IV. 시뮬레이션 결과 분석

제안된 알고리즘의 분석을 위해 ns-2 시뮬레이터[7]를 사용하였다. 기존의 RIO 알고리즘에 제안 알고리즘을 추가하고, 2장에서 보여준 시뮬레이션 구성을 그대로 사용하였다.

표 3. RIO와 새로운 RIO 알고리즘 비교

| flow | Target rate | RIO | New RIO |
|-----------------|-------------|----------|-----------|
| 1 | 0.1M | 0.269161 | 0.2374438 |
| 2 | 0.1M | 0.251354 | 0.254912 |
| 3 | 0.2M | 0.348145 | 0.3655942 |
| 4 | 0.2M | 0.363411 | 0.339618 |
| 5 | 0.5M | 0.643317 | 0.6009074 |
| 6 | 0.5M | 0.627072 | 0.61648 |
| 7 | 1M | 1.062987 | 1.0545166 |
| 8 | 1M | 1.032455 | 1.066432 |
| 9 | 2M | 1.812484 | 1.8764648 |
| 10 | 2M | 1.835279 | 1.8764088 |
| Target Achieved | | 8.245666 | 8.2887156 |
| BE Achieved | | 1.531294 | 1.4348994 |

표 3은 위의 시뮬레이션 구성표를 그대로 적용하여 확장된 RIO와 기존의 RIO를 적용시 각 흐름당 전송률을 비교한 것이다. 기존의 RIO에 비해 제안한 새로운 RIO 알고리즘은 높은 목표 전송률을 가진 흐름들이, 최선형 서비스 또는 낮은 목표 전송률을 가진 흐름들에 대해 자신의 목표 전송률에 근접함을 알 수 있다. 여기서 상수 k 값은 이상적인 C 값을 말한다. 여기서 k 값은 시뮬레이션을 통해 얻은 평균값을 적용한 것이다.

기존의 RIO가 가지는 편차와 개선된 RIO의 편차가 크지 않지만, 높은 목표 전송률을 가진 흐름들이 자신들의 목표 전송률에 근접함을 알 수 있다. 그림 6에서 기존의 RIO와 확장된 RIO의 전송률의 차이를 보여준다.

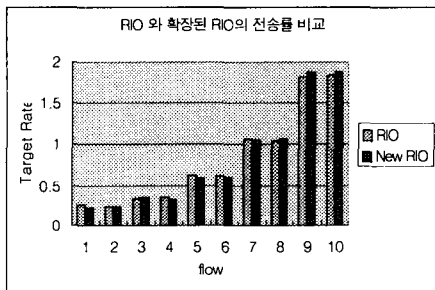


그림 6. RIO와 확장된 RIO의 전송률 비교

결과값의 차이가 크게 나지 않는 이유는 Ibanez와 Nichols[1], 그리고 염익준[3]과 박원형[2]이 공통적으로 지적했듯이, TCP 메커니즘에 원인이 있다고 할 수 있다. 흐름들의 각각의 윈도우를 W라고 할 때, 한 패킷이 폐기되면 TCP 메커니즘에서는 강 혼잡 회피를 위해 W 사이즈를 1/2W로 줄

인다. 각각의 흐름들은 W를 원상태로 돌리는데 걸리는 시간이 각각의 윈도우 사이즈 크기에 따라서 다르기 때문에, 자신의 전송률을 복구하는 것은 W가 작은 흐름이 유리할 수 밖에 없다. 결과적으로 TCP 메커니즘이 개선이 되지 않는 한 대역폭 스쿠 문제는 계속적으로 발생한다.

V. 결론

인터넷 사용자들의 다양한 서비스 요구를 수용하기 위해 IETF는 차별 서비스 네트워크 모델을 제안했고, RIO는 차별서비스를 구현하기 위한 중요한 큐 관리 기법으로 자리잡았다. RIO는 사용자들이 요구하는 QoS를 보장 할 수 있는 가장 적절한 메커니즘이라고 할 수 있다. 대역폭 스쿠 문제는 RIO를 구현시에 가장 우선적으로 해결해야 할 문제다. 본 논문에서는 이러한 목표 전송률의 값에 따른 영향을 완화하기 위한 확장된 RIO를 제안하였다. 기본적인 메커니즘은 RIO에 기반하고, 큐 내의 IN 패킷의 점유율에 따라 OUT 패킷의 폐기 확률을 증가 또는 감소시킴으로써 대역폭 스쿠 문제를 완화시킨다. ns-2 시뮬레이터를 사용하여 확장된 RIO의 성능을 분석한 결과, 대역폭 스쿠 문제를 상당 부분 완화시키는 결과가 나왔으나, 완전히 제거하지는 못한다. 이것은 TCP 혼잡 제어 메커니즘이 갖는 한계성에 크게 기인한다. 앞으로 다양한 네트워크 구조를 가지고 성능평가를 분석하고, Round Trip Time 과 패킷 크기에 따른 영향을 분석하여 RIO를 보완하는 연구가 진행될 것이다.

참고문헌

- [1] J. Ibanez and K. Nichols, "Preliminary Simulation Evaluation of an Assured Service", draft-ibanez-diffserv-assured-eval-00.txt, Internet Draft, August 1998.
- [2] W. H. Park and H. Kim, "A modified RIO Algorithm that alleviates the Bandwidth Skew problem in Internet Differentiated Service"
- [3] I. Yeom and A. L. N. Reddy, "Realizing Throughput Guarantees in Diff-Serv Networks", IEEE ICMCS 99.
- [4] S. Blake, et al, "An Architecture for Differentiated Service" RFC2475, Dec. 1998
- [5] D. Clark and Fang W, "Explicit Allocation of Best Effort Delivery Service", IEEE/ACM Transactions on Networking 6, 4(1998)
- [6] S. Floyd and V. Jacobson, "Random Early Detection (RED) gateways for: Congestion Avoidance", IEEE/ACM Transactions on Networking, 1(4), Aug. 1993, pp. 397-413
- [7] UCB/LBNL/VINT Network Simulator - ns2, URL : www-mash.cs.berkeley.edu/ns/.