濟州大學校 尖端技術硏究所 論文集, 第13卷, 2號, pp.86-92, 2002. 12
Cheju Nat'l Univ. Res. Inst. Adv. Tech. Jour., Vol.13, No.2, pp.86-92, 2002. 12

TCP 혼잡 제어를 위한 동적 RED 알고리즘

김 대 영* · 김 은 범* · 안 기 중**

Dynamic RED Algorithm for TCP congestion control

Dae-Young Kim*, Eun-Bum Kim* and Khi-Jung Ahn**

ABSTRACT

Recently many studies have been done on the Random Early Detection as an active queue management and congestion avoidance scheme in the Internet. The AQM(Active Queue Management) of traffic congestion involves the policy which drops a packet for congestion avoidance. In this case, router drops one or more packets to improve the network performance before when the output buffer becomes full. In this paper, we overview the characteristics of the RED and the modified RED algorithms in order to understand the current status of these studies. Based on the RED parameter analysis, we propose a Dynamic RED Algorithm for Active Queue Management scheme to cope with this RED weakness. In this algorithm we make three parameters be adjusted depending on the queue status.

Key Words : Traffic congestion, AQM, RED, congestion avoidance

Ⅰ. 서 론

TCP 종단대 종단 혼잡제어 메커니즘은 인터넷 망에서 중요한 요소이다. 인터넷 사용자의 급증과 다양한 인터넷 기반 응용프로그램의 등장으로 인하여 인터넷망은 자체적으로 자신의 자원을 제어할 수 있어야 한다. 인터넷 사용자의 폭발적 증가 즉, 채팅과 같은 소량의 데이터의 서비스는 물론 비디오와 같은 대량의 데이터 서비스 등이 확대가 되고, 이로 인하여 대역폭의 결핍을 가져오게 될 것이고, best-effort 트래픽을 서비스하기 위하여 게이트웨이(Router)에서 혼잡제어가 이루어 져야 한다[1].

전송 중에 발생하는 패킷 손실의 원인을 네트워크 혼잡에 두고 이를 줄이기 위한 방향으로 개선되어 온 TCP 프로토콜은 통신의 양 종단간 신뢰성 있는 데이터 전송을 보장하며 인터넷에서 가장 보편적으로 쓰인다.

TCP 통신망에서의 흐름제어는 크게 두 가지로 나눌 수 있다. 첫째, 종단간 혼잡제어(end-to-end congestion control)이다. 즉, TCP connection의 종단간, 전송 측과 수신 측에서 패킷을 제어하는 것을 의미한다.

두 번째, 게이트웨이 혼잡 제어를 들 수 있다. 이는 단말 혼잡 제어를 보완하기 위해서 통신망의 게이트웨이 상에 혼잡 제어 기법을 구현하는 방법이다.

TCP망에서 혼잡제어를 위한 방법으로 중간시스템

* 제주대학교 대학원
Graduate School, Cheju Nat'l Univ.
** 제주대학교 통신 컴퓨터 공학부, 첨단 기술연구소
Faculty of Telecommunication and Computer Engineering, Res. Insti. Adv. Tech., Cheju Nat'l Univ.

인 라우터에서 혼잡 예측을 통한 혼잡회피 방법이다.

AQM(active queue management)은 라우터 큐에서의 혼잡상태 정보를 단말 호스트에 전달하고, 전송속도를 제어할 수 있도록 하여 패킷 손실율과 전송지연을 줄여서 TCP에서 혼잡을 회피하는 방법으로 널리 이용하고 있다. 이에 대한 대표적인 알고리즘으로서 RED가 있다[2]. 이에 대한 변형 알고리즘으로 Stabilized RED[3], Adaptive RED[4], Flow RED[5], BLUE[6], AVQ(adaptive virtual queue)[7], REM (random exponential marking)[8], CHOKe(CHOose and keep)[9] 등이 있다.

TCP에서 혼잡회피를 위한 AQM 알고리즘은 큐 가중치와 최대 폐기확률을 조정함으로서 다양한 트래픽 상황에 따라 평균 큐 길이를 조정하는 것이 지금까지 연구 결과이다[1][10][11]. 그러나 이러한 알고리즘은 경험적 파라메터을 적용함으로서 다양한 트래픽에 능동적으로 적용하기가 곤란하다.

따라서 본 논문에서는 고정적 파라메터를 사용할 경우에 다양한 트래픽 상황에 적절하게 대응하지 못하는 문제를 해결하기 위하여 파라메터 설정을 동적으로 설정함으로서 라우터의 버퍼관리를 능동적 관리를 할 수 있도록 하는 동적 RED 알고리즘(Dynamic RED Algorithm)을 제안하고자 한다.

버퍼관리를 위한 파라메터로서 최대 확률, 평균 큐 가중치의 값, 버퍼의 최소임계값, 최대임계값을 동적으로 조정하고자 한다.

2장에서는 TCP/IP 망에서의 트래픽 제어기술로 지금까지 연구되었던 AQM(active queue management)에 대한 파라메터 설정에 대한 것을 기술하고, 3장에서는 본 본문에서 제안한 동적 RED 알고리즘(Dynamic RED Algorithm)을 소개한다. 4장에서는 시뮬레이션 및 성능분석으로 제안 알고리즘의 모의 실험결과를 제시하며, 5장에서는 요약 및 결론을 맺는다.

II. AQM 관련 연구 및 특징

AQM(active queue management)은 라우터 큐에서의 혼잡상태 정보를 단말 호스트에 전달하고, 전송속도를 제어할 수 있도록 하여 패킷 손실율과 전송지연을 줄여서 TCP에서 혼잡을 회피하는 방법으로 널리 이용하고 있다. 즉, 평균 큐 길이에 따라 망의 체증 정도를 결정하며 이에 따라 패킷을 마크 혹은 폐기시키는 확률을 결정한다. 한편 송신 호스트는 해당 패킷의 처리 여부에 따라 전송속도를 조절한다.

2.1. RED

RED는 도착하는 패킷에 대하여 라우터의 큐 상태에 따라 선택적 폐기를 통하여 망의 혼잡상태로 이르는 것을 방지하는 알고리즘으로서 새로운 패킷이 도착할 때마다 큐가 비어있는지의 여부를 파악하기 위하여 EWMA에 의해 평균화된 큐의 길이를 선형적 함수에 적용하여 확률적으로 패킷을 폐기한다[2].

$$Q_{avg} \leftarrow (1-w_q) * Q_{avg} + w_q * Q_{size} \quad (1)$$

(Q_{avg} : average queue size, w_q : queue weight
Q_{size} : current queue size)

식(1)을 미리 정해 놓은 큐 임계값($\min_{th}$)과 최대 큐 임계값($\max_{th}$)과 비교하여 Q_{avg}가 $\min_{th}$ 보다 작을 때에는 모든 패킷은 정상적으로 처리되고 $\max_{th}$ 보다 클 때에는 마크되거나 폐기된다. 그리고 Q_{avg}가 $\min_{th}$와 $\max_{th}$ 사이에 있을 때는 도착하는 패킷은 폐기확률 P_b을 적용 받는다.

$$P_b = P_{\max} \times \frac{Q_{avg} - \max_{th}}{\max_{th} - \min_{th}} \quad (2)$$

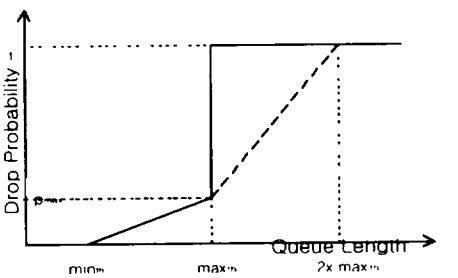

Fig. 1. Drop Probability.

이와 같은 RED의 확률적 폐기는 TCP의 AIMD (additive increase/multiplicative decrease)알고리즘[12][13]과 함께 사용되었을 경우 라우터에서 전역동

기화문제를 피할 수가 있어 버스트한 트래픽에 의한 영향을 줄일 수 있다. 그러나 RED의 문제점은 라우터에서 공유하고 있는 링크에 대하여 트래픽 형태에 구분 없이 동일한 기준의 폐기확률을 적용하고 있어 네트워크 환경 설정이 어렵다[1][11].

이를 극복하여 RED의 성능을 향상시키기 위한 새로운 형태의 AQM들을 제안하고 있다.

2.2. FRED

FRED(Flow RED)는 트래픽 흐름에 순응하도록 하기 위한 것으로서 트래픽 유형에 상관없이 공정한 링크 사용이나, 큐 버퍼 사용에 제한적이므로 대형 버퍼를 갖는 라우터에서 효과적인 제어 방식이다[5].

2.3 Adaptive RED

Adaptive RED는 RED의 폐기를 위해 ECN 마크를 사용하는 최대 확률 P_{max}의 고정된 값을 조정함으로서 더 높은 링크 효율을 얻을 수 있다[4]. 이것은 큐 길이를 변화를 추적하여 최대 확률을 적용함으로서 더 높은 링크 효율을 나타내고 있다. 그러나 Adaptive RED 알고리즘에서 적용되는 파라메터들의 변화가 기존 RED의 파라메터 초기 값을 변형한 것과 큰 차이를 보이고 있지 않다[1]. 이것은 RED에서 초기 파라메터 값을 어떻게 설정하느냐에 따라 RED와 같은 성능을 보여준다는 것을 의미한다.

2.4. Blue

Blue는 큐의 범람이 발생하여 패킷을 버리게 되었을 경우 폐기 확률을 일정 값만큼 줄이고, 언더플로우가 발생하여 큐가 비어 있을 경우 폐기확률을 일정 값만큼 증가시킨다. 단, 큐의 변화가 너무 빨리 일어나지 않도록 Freeze time 동안에는 범람이나 언더플로우가 발생하더라도 폐기 확률을 변화시키지 않는다[6].

2.5. REM

REM(random exponential marking)은 지역 시스템의 안정성을 위해 도입한 PI(proportional-integral) 제어기에 이용한 값을 지수함수에 적용하여 폐기 확률을 구하는 방법이다[8].

$$p = a*(Q_{size} - Q_{ref}) - b*(Q_{old} - Q_{ref}) + p_{old} \quad (3)$$

$$prob = 1 - \phi^{-p} \quad (\phi > 1) \quad (4)$$

REM 알고리즘은 패킷이 여러 개의 혼잡상태에 있는 라우터를 지나갈 때 각각의 라우터에서 겪게 되는 확률을 고려하면 폐기 확률은 다음과 같이 주어진다.

$$prob = 1 - \phi^{\sum P_i} \quad (5)$$

2.6. AVQ

AVQ(adaptive virtual queue)도 지역 안정성을 위하여 PI 제어기와 마찬가지로 TCP 혼잡제어 알고리즘을 선형화한 것으로서 패킷이 도착할 때마다 가상큐(virtual queue)에 패킷을 더하고 Capacity를 다시 계산하여 가상큐에 overflow 발생하면 폐기한다[7].

2.7 AQM 알고리즘의 특징

AQM의 개선 알고리즘들은 RED을 바탕으로 다양한 트래픽 환경에 적용할 수 있도록 적은 손실, 지연의 최소화, 링크 효율을 극대화하면서 공정성을 제공하고자 하는 것이다. 따라서 큐 관리의 가장 큰 관점은 평균 큐 길이의 가중치와 링크 효율을 극대화시키는 최대 확률을 조정하는 것이라 할 수 있다.

III. Dynamic RED Algorithm

3.1. RED 파라메터의 특징 및 문제점

RED 알고리즘에서는 평균 큐 길이를 이용하여 도착하는 패킷에 대하여 랜덤하게 폐기 확률을 적용함으로서 망의 혼잡을 회피한다. 그러나 평균 큐 길이를 이용함으로서 다음과 같은 문제를 야기 시킨다.

- 평균 큐 길이를 구할 때 실제 큐 길이가 Q_{avg}에 변하는데 미치는 영향이 w_q값이 작을수록 작다. 이러한 경우 순간적인 기간동안 큐의 길이가 증

가하더라도 Q_{avg}에 영향이 적기 때문에 일시적인 트래픽에 폐기확률을 적용할 수가 없다.

- 전역 동기화가 발생하여 현재 큐가 비어 있음에도 불구하고 Q_{avg}가 $\max_{th}$ 근접해 있어 도착하는 패킷에 대해 확률 P_a가 적용되어 폐기 처리하게 되어 재전송으로 인한 지연이 발생한다.
- 또한 w_q값을 크게 할 경우 Q_{avg}에 영향을 크게 반영되므로 Q_{avg} 변화가 심하므로 라우터의 버퍼 관리를 효율적으로 수행하기가 어렵고 Drop-Tail 과 같은 효과를 나타낸다.

또한 최대 확률 $P_{\max}$는 Q_{avg}에 영향을 주게 되는데 폐기 확률을 적용함에 있어서 선형적으로 증가하므로 다음과 같은 문제점이 있다.

- $P_{\max}$값이 클수록 폐기 확률이 커지므로 폐기되는 횟수가 많아지게 되어 링크 효율이 떨어지게 된다.
- $P_{\max}$값이 작을수록 폐기 확률이 적어지므로 Q_{avg} 변화가 적게 되므로 체증이 발생하여 전역 동기화가 자주 발생할 수 있다.
- 고정적인 $P_{\max}$값으로 Q_{avg}을 조절하지 못하는 버스트한 트래픽이 발생하면 Q_{avg}가 $\max_{th}$값 보다 떨어질 때까지 계속해서 폐기를 하게 되므로 결국 전역동기화 발생한다.

그리고 버퍼관리를 위한 임계값($\min_{th}$, $\max_{th}$)은 트래픽이 버스트한 특성을 가질수록 $\max_{th}$을 크게 주어 링크 효율을 높게 하지만, 전체 큐 길이가 길어지게 되므로 링크 효율과 전송지연은 절충관계가 된다. 효과적인 $\max_{th}$값은 $\min_{th}$의 두 배 이상 사용할 것을 권고하고 있다[2].

3.2. Dynamic RED 파라메터 설정

위에서 언급한 파라메터의 문제점을 해결하기 위해 평균 큐 길이의 변화에 작용하는 평균 큐 길이 계산 가중치 w_q와 $P_{\max}$, $\max_{th}$를 동적으로 할당하는 알고리즘을 제안하고자 한다.

1) 가중치 w_q 파라메터 설정

다양한 회선 대역폭(link bandwidth)에 적용하기 위하여 w_q를 설정하는데 Link Capacity 값을 적용하여 계산한다.

$$w_q = 1 - \exp(-1/C) \tag{6}$$

또한 평균 큐 길이가 $\min_{th}$와 $\max_{th}$사이 있을 경우는 w_q값을 감소시키고, 평균 큐 길이가 $\max_{th}$ 보다 큰 경우는 w_q값을 증가시켜 평균 큐 길이가 빠르게 $\max_{th}$이하로 떨어지게 하여 전역동기화 후에도 들어오는 패킷에 대하여 폐기를 하지 않도록 하였다.

$$\begin{aligned} w_q &= w_q - \alpha \\ w_q &= w_q + \beta \end{aligned} \tag{7}$$

(α : Increase value 0.0001,
β : Decrease value 0.0002)

2) 최대 확률 $P_{\max}$ 설정

RED에서 체증회피를 의한 폐기 확률은 평균 큐 길이에 따라서 0에서부터 $P_{\max}$까지 범위를 갖는 확률로서 평균 큐 길이에 따라서 폐기함으로써 체증 발생 전에 큐에 들어오는 것을 막아 큐가 안정상태가 되도록 하므로 평균 큐 길이 변화에 중요한 요소이다. 체증 판단은 평균 큐 길이가 최대 임계값을 넘었을 경우이므로 이때 최대 확률 $P_{\max}$을 증가시켜줌으로써 게이트웨이에서 들어오는 패킷에 대해 많게 폐기시킴으로서 혼잡회피를 능동적으로 대처할 수 있다.

Adaptive RED에서는 최대 확률 $P_{\max}$변화는 천천히 그리고 드물게 조정되어야 하며, $P_{\max}$의 범위는 값은 0.01과 0.5사이의 값으로 제한하는 것을 권장하고 있다[2][4].

본 논문에서는 $P_{\max}$값을 조정함에 있어서 AIMD 방법을 사용하여 $P_{\max}$값을 조정하였고, 조정하기 위한 요소로서 평균 큐 길이에 대한 목적 큐 길이라는 것을 설정하였다.

$$TQ_{avg} = Range[\, TH_{\min} + 0.4*(TH_{\max} - TH_{\min}),\; TH_{\min} + 0.6*(TH_{\max} - TH_{\min})] \tag{8}$$

$$P_{\max} = P_{\max} + \alpha \quad P_{\max} = P_{\max} * \beta \tag{9}$$

(α: $MIN(0.01,\ P_{\max}/4)$ β: 0.8)

3) 임계값 $\min_{th}$와 $\max_{th}$ 설정

임계값 설정은 Link Capacity에 따라 지연에 밀접한 관계를 갖고 있으므로 Link 효율과 전송지연의 절충관계에 놓이게 되고, 다양한 Link Capacity에 적용하기 위하여 다음과 같은 식을 적용하였다.

$$\min = MAX\left[\frac{1}{8}Q_{\max}, D_{\sec/2}\right]$$
$$\max_{th} = 4 * \min_{th} \quad (10)$$

($D_{\sec}$: Target average queuing delay second for Link Capacity C)

시뮬레이션에서는 최소 임계 값을 20으로 설정하여 성능 측정을 하였다.

3.3. Dynamic RED algorithm

파라메터(w_q, $P_{\max}$, $\min_{th}$, $\max_{th}$)값은 다양한 트래픽 환경에 동적으로 적응할 수 있는 알고리즘은 Fig. 2와 같다. 기존의 RED 알고리즘에서 파라메터 조정에 대한 부분을 추가하였다.

Initialization :
$Q_{avg} \leftarrow 0$
$count \leftarrow -1$
$w_q \leftarrow 1 - e^{(-1/C)}$
$\min_{th} \leftarrow MAX\left[\frac{1}{8}Q_{\max}, D_{\sec}/2\right]$
$\max_{th} \leftarrow 4 * \min_{th}$
For each packet arrival
Calculate the new average queue size Q_{size}
if $Q_{size} > 0$ then // queue non-empty
$Q_{avg} \leftarrow (1-w_q) * Q_{avg} + w_q * Q_{size}$
else
$m \leftarrow f * (time - q_{time})$
$Q_{avg} \leftarrow (1 - w_q)^m * Q_{avg}$
end if
$TQ_{avg} \leftarrow Range[(\min_{th} + 0.4*(\max_{th} - \min_{th}), \min_{th} + 0.6*(\max_{th} - \min_{th})]$
if $\min_{th} \le Q_{avg} < \max_{th}$ then
increment *count* //(w_q, $P_{\max}$)값 조정시작
$w_q \leftarrow w_q - 0.001$ // w_q 감소
if ($Q_{avg} > TQ_{avg}$ and $P_{\max} \le 0.5$) //확률조정
$\alpha \leftarrow MIN(0.01, P_{\max}/4)$ //증가치 계산
$P_{\max} \leftarrow P_{\max} + \alpha$ // $P_{\max}$ 증가
else if ($Q_{avg} < TQ_{avg}$ and $P_{\max} \ge 0.01$)
$P_{\max} \leftarrow P_{\max} * \beta$ // $P_{\max}$ 감소
endif
calculate probability P_a :
$P_b \leftarrow P_{\max}(Q_{avg} - \min_{th})(\max_{th} - \min_{th})$
$P_a \leftarrow P_b/(1 - count \times P_b)$
with probability P_a
Mark the arriving Packet
$count \leftarrow 0$
else if $Q_{avg} \ge \max_{th}$ then
$w_q \leftarrow w_q + 0.002$ // w_q 증가
$\alpha \leftarrow MIN(0.01, P_{\max}/4)$ // 증가치 계산
$P_{\max} \leftarrow P_{\max} + \alpha$ // $P_{\max}$ 증가
//(w_q, $P_{\max}$) 값 조정 끝
else
Queue Packet
$count \leftarrow -1$
When queue becomes empty
$q_{time} \leftarrow time$

Fig. 2. Dynamic RED Algorithm.

Ⅳ. 시뮬레이션 및 성능분석 결과

4.1 시뮬레이션 환경

본 논문에서 사용한 네트워크 환경은 Fig. 3과 같이 구성하였으며, 다음과 같은 전제 조건을 두었으며

시뮬레이션은 $\max_{th}$과 $\min_{th}$값을 초기 값을 설정하기 위하여 link Capacity를 100Mbps, target queuing delay time을 1에서부터 5ms까지를 변화시키면서 적용하였으며, RED에서 사용하는 파라메터를 동적으로 조정하기 위하여 w_q값과 $P_{\max}$값을 변화는 과정을 살펴보았다.

최대 확률 $P_{\max}$는 $MIN(0.01, P_{\max}/4)$를 이용하여 증가시켰으며, 감소하는 Factor 0.8을 사용하였다.

w_q값은 평균 큐 길이 변화에 너무 민감하게 변하지 않도록 하기 위하여 증가치는 0.001로 설정하였고 감소치는 평균 큐 길이가 빠르게 $\max_{th}$이하로 떨어

지게 하기 위하여 0.002를 적용하였다.

시뮬레이션은 링크 수를 늘여가면서 Gateway 형태를 RED. Dynamic RED의 방식을 비교하였으며. 시뮬레이션 시간은 10초. Gateway 큐 길이는 100패킷. 1패킷은 1000byte로 하였고 Ack packet은 40byte로 하였다.

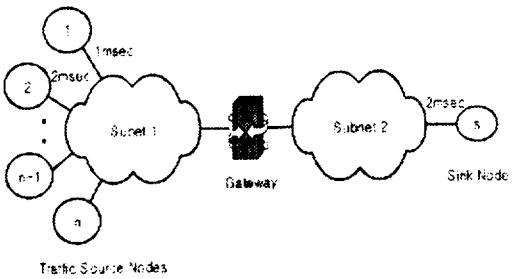

Fig. 3. Simulation Networks.

4.2 시뮬레이션 결과

파라메터를 자동 조정하게 하는 것은 Link Utilization과 전송지연과는 절충관계의 결과를 나타내고 있고. w_q값과 P_{max}의 조정으로 인한 체증 회피로 인하여 패킷 손실율이 증가함을 보여주었다. 반면 평균 큐 길이는 RED보다 더 큰 값을 유지하였으며. 다양한 트래픽에 대한 라우터에서의 능동적인 큐 관리 기법이라 하겠다.

Fig. 4와 Fig. 5에서 보여주고 있는 것은 DRED가 RED에서 비해서 평균 큐 길이가 라우터에 도착하는 실제 큐 길이 변화에 빨리 적응하고 있어 체증 발생이 적음을 보여주고 있고 전역 동기화가 덜 발생하고 있다.

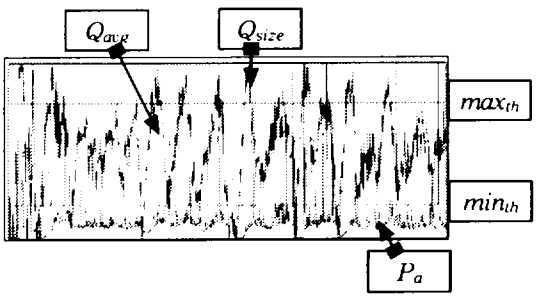

Fig. 4. Performance of RED.

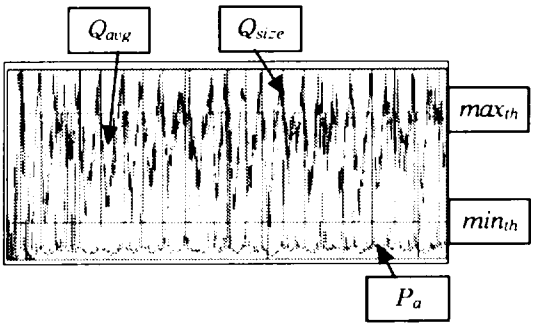

Fig. 5. Performance of DRED.

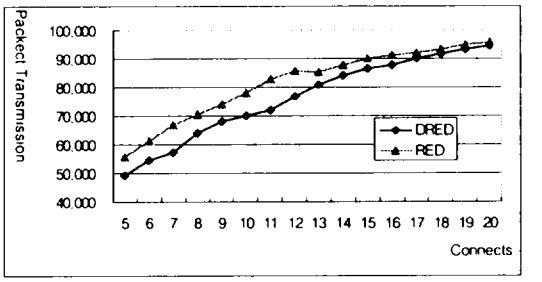

Fig. 6. Comparison of Packet Transmissions.

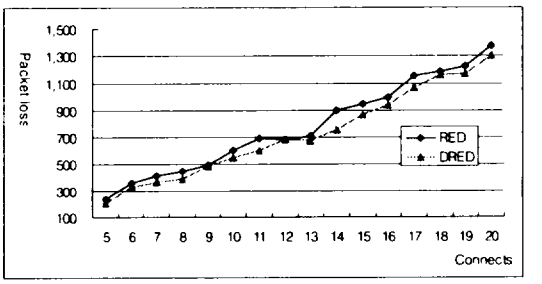

Fig. 7. Comparison of Packet Losses.

Fig. 6에서 보여주고 있는 것은 패킷 전송은 폐기확률의 증감으로 인하여 DRED가 더 많은 패킷을 폐기함으로서 전송지연이 발생하고 있어 이로 인한 Link 효율을 떨어지고 있다.

Fig. 7에서 보여 주고 있는 것은 패킷 손실율은 체증 발생이 적음으로 인하여 DRED가 RED보다 좋은 성능을 보여주고 있다. 즉 도착하는 패킷에 대한 평균 큐 길이 적응이 빠르므로 셀 손실율이 DRED가 더 좋은 성능을 보여주고 있음(전역 동기화 문제 해결).

V. 결 론

능동적인 큐 관리 위한 기법이 여러 가지 제안되고 있으나 네트워크 환경의 다양성 및 이용자들이 다양한 서비스를 인터넷 망을 통해서 원활하게 서비스하기 위해서는 데이터 전송은 중요하다. 망의 체증 상태를 조기 감지하여 능동적으로 큐 관리를 위한 TCP 종단간 혼잡제어는 종단 노드가 패킷 손실이나 응답 시간의 변동에 의해 간접적으로 네트워크의 부하를 추정하고 그것에 따라서 혼잡 제어를 한다. 하지만 종단 노드만으로 발생한 혼잡을 해소하는 것은 쉬운 일이 아니며 종단 노드에서의 혼잡 제어 방안들이 중요시되고 있다.

본 논문에서 고정적인 파라메터를 적용하는 대신 망 상태에 따라 파라메터를 동적으로 관리하는 변화시켜보았다. 연구 결과 모든 트래픽 상황에 맞는 최적의 파라메터를 얻는 것은 전송지연과 처리율과는 절충관계에 있다.

향후 연구과제로서는 동적으로 파라메터를 적용하면서 전송지연을 최소화하고 처리량을 극대화시키는 파라메터 분석이 필요하다.

참고문헌

1) Sally Floyd. Ramakrishna Gummadi. and Scott Shenker. August 2001. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management.

2) Floyd. S.. and Jacobson. V. August 1993. Random Early Detection gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking. Vol.1 No.4. p. 397-413

3) T. J. Ott. T. V. Lakshman. L. H. Wong. March 1999. SRED: Stabilized RED. INFOCOM.

4) W. Feng. D. Kandlur. D. Saha. K. Shin. March 1999. A Self-Configuring RED Gateway. INFOCOM.

5) D. Lin and R. Morris. 1997. Dynamics of Random Early Detection. Proc. ACM SIGCOMM.

6) W. Feng. D. Kandlur. D. Saha. K. Shin. April 1999. Blue: A New Class of Active Queue Management Algorithms. U. Michigan CSE-TR-387-99.

7) S. Kunniyur. R. Srikant. August 2001. Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management. SIGCOMM.

8) S. Athuraliya. V. Li. S. Low. Q. Yin. May 2001. REM: Active Queue Management. IEEE Network.

9) R. Pan et al.. 2000. CHOKe: A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation. Proc. IEEE Infocom.

10) 유영석. 홍석원. 1997. 11. 체증 제어를 위한 Random Early Detection(RED) 알고리즘의 파라미터 분석. 통신학회 추계발표 논문집 p.41-44.

11) 김대영. 김동춘. 안기중. 2002. 4. 큐의 변화량에 의한 예측기반 RED 알고리즘". JCCI 2002. Section III-D.2.1~4

12) V. Jacobson. August 1998. "Congestion Avoidance and Control". SIGCOMM.

13) W. Stevens. 1994. "TCP/IP Illustrated". Vol.1. Addison-Wesley Publishing company.

14) 김복심. 2001. 12. 예측기반 RED알고리즘. 제주대학교 논문집.