인터넷에서 혼잡제어를 위한 개선된 RED 알고리즘

정 규 정† · 이 동 호††

요 약

혼잡상황이 발생한 후에는 네트웍 성능이 급격하게 떨어지게 되는 문제점을 해결하고자 큐 관리 알고리즘으로 RED(Random Early Detection) 알고리즘이 제시되었으며, 현재 RED 알고리즘이 표준으로 사용되고 있다. RED 알고리즘은 게이트웨이에서 능동적으로 대처할 수 있는 알고리즘이다. 그러나, RED 매개변수 분석을 통하여 고정된 매개변수 값을 사용할 경우, 현재 네트웍 트래픽 상태에 적절히 대처하지 못하는 문제점을 가지고 있다. 본 논문에서는 기존의 RED 알고리즘의 문제점인 고정된 매개변수 값 설정으로 인한 문제점을 해결하기 위하여 개선된 RED 알고리즘을 제안한다. 제안된 알고리즘에서는 현재의 네트웍 트래픽 상태에 따라 패킷 폐기 확률 값을 조정하여 게이트웨이에서의 혼잡상황을 효율적으로 제어하도록 하였다. 그리고, NS를 이용하여 개선된 RED 알고리즘의 성능을 검증하였다.

An Effective RED Algorithm for Congestion Control in the Internet

Kyu-jung Jung† · Dong-ho Lee††

ABSTRACT

The network performance gets down during congestion periods to solve the problem effectively. A RED(Random Early Detection) algorithm of the queue management algorithm is proposed and IETF recommends it as a queue management. A RED algorithm controls a congestion aspect dynamically. In analyzing parameters when static value of parameter is set in the gateway cannot be handled the status of current network traffic properly. We propose the Effective RED algorithm to solve with the weakness of RED. In this algorithm the maximum drop probability decides to accept or drop the incoming packets, is adjusted dynamically on the current queue state for controlling the congestion phase effectively in the gateway. This algorithm is confirmed by computer simulation using the NS (Network Simulator)-2.

키워드 : RED, 혼잡제어(Congestion Control)

1. 서 론

현재 인터넷은 Best-Effort 방식의 패킷 스위칭 네트워크에서는 가능한 한 패킷을 전송하려고 하기 때문에 갑자기 폭주하는 트래픽은 일정한 전송품질을 보장받지 못하게 되고 혼잡문제가 발생한다. 이러한 혼잡을 제어하기 위해서는 네트워크의 혼잡상태를 송신측에 알려주는 피드백이 필요하다. 송신측은 이러한 혼잡 신호를 받고 네트워크로 내보내는 패킷을 줄여야 하는데 이러한 전송율 조정이 혼잡제어의 기본개념이다[1]. 인터넷에서의 혼잡제어 분야에서 많은 연구가 이루어졌다. 네트워크의 규모가 점점 커지고 다양한 시스템이 연결됨에 따라 현재 인터넷은 끊임없는 변화에 직면하고 있으며 혼잡제어가 없는 UDP를 사용하면서 멀티캐스트 방식으로 실시간으로 전송되는 멀티미디어 트래픽이 날로 증가하고 있다. 현재 멀티캐스트 트래픽은 대부분 실시간 비디오/오디오 스트리밍 서비스와 같이 높은 대역폭을 필요로 하며 장시간 지속되며 흐름제어가 없는 특징 때문에 멀티캐스트 트래픽의 증가는 심각한 혼잡문제를 야기하게 된다. 현재 인터넷에서 TCP는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 방식의 혼잡제어 기법을 가지고 있다[2, 3]. 이러한 방법은 현재 네트워크 상의 상태를 파악하여 최대의 처리율, 적은 지연과 높은 속도를 지원하고자 고안된 것이다.

이러한 방법 또한 기존의 방법을 개선시켰지만 혼잡상황이 감지된 이후에는 전역 동기화 및 패킷 손실로 인하여 네트워크 성능이 급격히 떨어지게 된다. 이런 문제점을 해결하고자 큐 관리 알고리즘으로 IETF에서는 RED(Random

※ 이 논문은 2000년도 광운대학교 교내 연구비 지원에 의해 연구되었음.
† 정 회 원 : 한화 S&C 컨설팅사업부
†† 종신회원 : 광운대학교 컴퓨터과학과 교수
논문접수 : 2002년 11월 23일, 심사완료 : 2003년 1월 28일

Early Detection) 알고리즘을 권고하고 있다[4-6]. RED 알고리즘은 혼잡 상황으로 발생되었던 문제점인 전역 동기화 및 패킷 손실율을 낮추는데 성공적이었다[7].

그러나, RED 알고리즘의 경우 네트워크 상황에 따라 적절한 매개 변수 값을 설정하지 않는다면 Drop Tail 방법보다 효율이 저하되고 또한 매개 변수 값을 설정하는데 있어서도 큰 어려움이 있다.

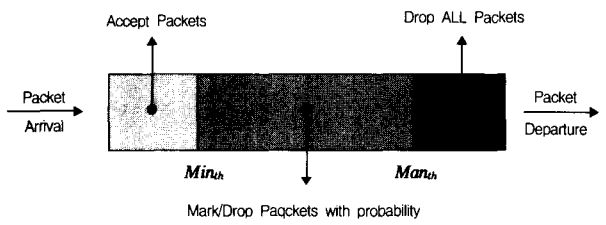
본 논문에서는 RED 알고리즘의 문제점으로 제기되고 있는 매개 변수 값을 현재 네트워크 상황에 맞추어 동적으로 변경하는 부분을 추가한 Effective RED 알고리즘을 제안하였다. 제안한 알고리즘은 기존의 RED 알고리즘보다 보다 효율적으로 큐를 관리하여 보다 좋은 성능을 제공한다.

2장에서는 현재 RED 알고리즘의 동작과 각 매개 변수 값에 대한 설명과 문제점에 대하여 설명하였다. 3장에서는 새롭게 제안한 Effective RED 알고리즘에 대하여 기술하였다. 4장에서는 시뮬레이터를 이용하여 제안한 알고리즘에 대한 검증을 하였다. 마지막으로 5장에서는 결론 및 향후과제를 제시하였다.

2. RED 알고리즘

2.1 기본 개념

RED 알고리즘은 버퍼에서의 평균 큐 크기(qave)를 이용하여 네트워크의 혼잡상황을 미리 감지하고 제어하는 기능을 가지고 있다[7]. RED는 혼잡상황을 전송 단말들에게 알려주는 방법으로서, 패킷의 ECN(Explicit Congestion Notification) 비트를 마킹하거나 혹은 폐기시키는 방법을 사용하고 있다. TCP를 이용하는 종단 단말에서는 패킷의 마크 필드 혹은 폐기를 확인하고, 혼잡 윈도우 값을 줄인후 슬로우 스타트를 실행한다[2, 6].



(그림 1) RED 큐의 상태와 동작 알고리즘

일반적으로, RED 알고리즘은 혼잡상황을 미리 감지하고 대처하기 때문에 네트워크 자원의 낭비를 줄여서 Drop Tail 알고리즘보다 우수한 성능을 나타낸다.

(그림 1)과 같이 패킷이 도착하게 되면, 평균 큐 크기(q_{ave}) < min_{th}이면 패킷을 큐에 넣고, 평균 큐 크기(q_{ave}) > max_{th}이면 패킷을 모두 폐기하고, min_{th} < 평균 큐 크기(q_{ave}) < max_{th}이면 랜덤하게 패킷을 마킹하거나 폐기하게 된다. 이 경우, 패킷이 폐기될 확률이 max_p값에 따라 변하게 되므로, 이 매개 변수 값의 설정이 RED 알고리즘에 큰 영향을 주게 된다. max_p값이 큰 경우에는 현재 유입된 패킷들이 마킹 혹은 폐기될 확률이 높아지게 되고, 반대로 max_p값이 작은 경우에는 현재 유입된 패킷들이 마킹 혹은 폐기될 확률이 낮아지게 된다.

RED 알고리즘은 <표 1>과 같이 다양한 매개 변수에 의해 제어되며, 그 중에서도 가장 큰 영향을 주는 것은 max_p이고, 이 값에 따라 전체적인 특성이 결정된다고 볼 수 있다. 그러나, 이렇게 다양한 매개 변수에 의해 동작되므로, 매개 변수 값이 잘못 설정된 경우에는 전체적인 성능이 Drop Tail 보다 떨어지게 된다[8].

w_q값은 평균 큐 크기(q_{ave})를 계산하는데 사용되는 가중치 값으로 이전의 평균 큐 크기가 현재의 평균 큐 크기에 어느 정도의 영향을 주는가를 결정하는 상수 값이다.

〈표 1〉 RED 매개 변수

$qlen$	최대로 수용 가능한 큐의 크기
min_{th}	폐기 여부를 결정하는 최소 임계값
max_{th}	폐기 여부를 결정하는 최대 임계값
w_q	q_{ave}(평균 큐 크기)를 계산하는 가중치 값
max_p	폐기 여부를 결정하는 최대 확률 값

$$q_{ave} = (1 - w_q) q_{ave} + w_q q_{ave}$$

위의 식과 같이, w_q값에 따라 현재 큐 크기의 비중을 높일 것인가 혹은 과거 큐 크기의 비중을 높일 것인지가 결정된다. w_q값을 결정하는 것도 중요한 것으로서, 이 값이 작은 경우에는 일시적인 큐 크기의 증가에 효율적으로 동작하지만, 게이트웨이에서 입력 패킷 모두를 폐기할 경우에는, 실제 큐 크기는 줄었으나 평균 큐 크기(q_{ave})가 감소되는 비율이 너무 느려 계속적으로 모든 패킷을 폐기하는 경우가 발생한다.

반대로, w_q값이 큰 경우에는 평균 큐 크기(q_{ave})도 빠르게 변화하지만, 일시적으로 폭주하는 패킷을 모두 수용하지 못하게 된다.

min_{th}, max_{th}값의 선택은 네트워크 특성에 따라 결정된다. min_{th}값이 너무 작게 되면, 랜덤하게 패킷을 마크 또는 폐기하는 구간이 크게 되어, 전체적인 네트워크 활용도가 떨어지게 된다. 반대로, min_{th}값이 너무 크게 되면, 빈번하게 혼잡 상황이 발생하게 된다. max_{th}값이 너무 작은 경우에도 큐의 사용가능한 크기가 작아져 네트워크 활용도가 떨어진다. 반대로, max_{th}값이 너무 크면, Drop Tail과 비슷한 동작을 하게 된다. 지금까지 서술한 바와 같이 RED 알고리즘에서는 매개 변수 값의 결정이 아주 중요하고, 그 값에 따라 RED 알고리즘의 성능이 크게 좌우되는 것을 알

수 있다. <표 2>에서는 RED 매개 변수들의 값을 설정하는 데 있어 각 매개 변수 값의 권고치가 나타나있다[9].

〈표 2〉 RED 매개 변수 값의 권고치

min_{th}	5 패킷
max_{th}	min_{th} 값의 3배
w_q	0.002
max_p	0.02~0.1

본 논문에서는 RED 알고리즘의 매개 변수 값중 가장 큰 역할을 하고 있는 max_p값의 설정을 현 네트워크 상황에 맞추어 설정하는 Effective 알고리즘을 제안한다.

3. Effective RED 알고리즘

기존의 RED 알고리즘의 문제점을 해결하고자 제안된 Effective RED 알고리즘은 유입되는 패킷의 양에 따라 변화하는 큐의 크기를 줄여 평형상태에 도달할 수 있도록 max_p값을 조정하는 알고리즘이다. 제안한 알고리즘은 (그림 2)와 같다.

```
if curq ≥ max_th
  d1 ← a1
  d2 ← a2
  d3 ← a3
  d4 ← a4

else if curq < min_th
  d1 ← b1
  d2 ← b2
  d3 ← b3
  d4 ← b4

else
  d1 ← c1
  d2 ← c2
  d3 ← c3
  d4 ← c4
```

```
if ((curq > oldq)
                && (curq > avg))
  max_p ← max_p + d1

else if ((curq > oldq)
                && (curq < avg))
  max_p ← max_p - d2

else if ((curq < oldq)
                && (curq > avg))
  max_p ← max_p + d3

else if ((curq < oldq)
                && (curq < avg))
  max_p ← max_p - d4
```

(그림 2) Effective RED 알고리즘

본 논문에서 제안하는 Effective RED 알고리즘의 동작은 큐에 패킷이 유입하게 되면, 현재 큐 크기와 이전 큐 크기를 비교하여 큐 크기의 진동차를 줄여 평형상태에 가까워지도록 max_p값을 조정하는 알고리즘이다. 즉, 지금의 트래픽이 얼마나 버스트한지를 파악하기 위하여 일단 현재 큐 길이가 max_{th}보다 큰 값인지, 아니면 min_{th}와 max_{th}의 중간 값인지를 비교하여 각각의 경우마다 서로 다른 값의 가중치를 준다. 그리고 현재 큐 길이(***curq***), 이전 큐 길이(***oldq***), 평균 큐 길이(***avg***)라는 세 가지 항목을 비교하여 이전의 트래픽이 버퍼에서 얼마나 처리되었는지를 파악하여 max_p를 조정한다.

큐 크기의 진동차가 크다는 것은 네트워크의 상황에 맞지 않는 max_p값이 현재 설정되어 있다고 볼 수 있다. Effective RED 알고리즘은 이러한 큐의 진동을 줄여 평형상태가 되도록, 즉 패킷 폐기율을 줄여주게 된다.

4. 실험 및 성능 평가

기존의 RED 알고리즘과 본 논문에서 새로 제안된 개선된 RED 알고리즘을 어느 정도의 성능 향상을 기대할 수 있는지를 알아보기 위한 모의 실험 결과를 통해 기존의 방식에서는 평형상태(Equilibrium point)에 절대로 도달하지 못하고 지속적으로 진동이 발생하는 것을 볼 수 있었다. 반면에 개선된 RED 알고리즘을 적용한 모의 실험 결과에서는 평형상태에 근접하여 적은 진동만이 발생하여 현재 네트워트래픽 상황에 따라 적절하게 동작함을 볼 수 있었다. 또한, 패킷 폐기율이 감소되었음을 볼 수 있었다.

RED 알고리즘에서 사용되는 매개변수에 대하여 고정된 값을 사용하는 것보다 네트웍 트래픽 상황에 능동적으로 조정하는 개선된 RED 알고리즘이 평형상태(Equilibrium point)에 근접한 범위내에서의 움직임만이 나타나고 있으며, 즉 진동차가 적은 그래프가 나타남은 물론, 패킷 폐기율은 기존의 RED 알고리즘보다 감소되어 큐 관리 알고리즘의 요구조건에 충족된다는 것을 보여주고 있다[4].

제안한 알고리즘의 성능을 분석하기 위하여 큐의 변동치가 얼마나 개선되었는지 살펴보았다. 네트워크 환경은 총 16개의 송신측이 전송하는 혼잡링크에서 큐의 변화를 모니터링 하였다. (그림 3)과 같이 기존의 RED 방법보다 큐 크기의 변동이 줄어들었다.

RED 알고리즘을 적용하여 모의 실험한 결과와 본 논문에서 제안된 개선된 RED(Effective RED) 알고리즘을 적용하여 모의 실험한 결과가 각각 (그림 3)과 (그림 4)에 나타나있다. 각각의 매개변수 설정값은 아래의 <표 3>과 같다.

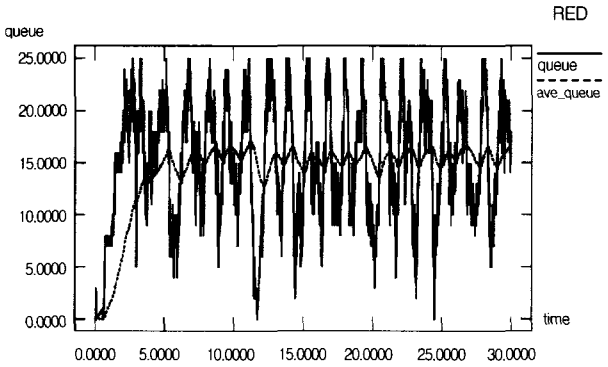
〈표 3〉 개선된 RED 모의실험의 매개변수 설정값

min_{th}	5 패킷
max_{th}	15 패킷
w_q	0.002
max_p	0.05
Node	송/수신 각 16개

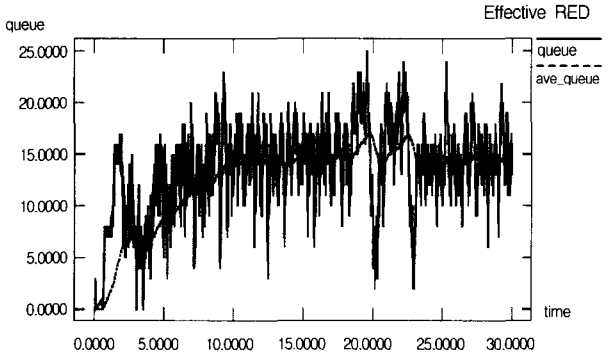
모의 실험 결과에서도 알 수 있듯이 max_p값을 잘못 설정하게 되면 (그림 3)에 나타나 있듯이 실제 큐 길이의 진동차가 심하게 나타나고, 이 진동차가 심하다는 것은 네트웍 성능에 좋지 않다고 할 수 있다[4]. 개선된 RED 알고리즘

을 적용한 (그림 4)를 보게되면 max_p값을 현재 트래픽 상황에 맞게 조절하여 평균 큐 길이를 중심점으로 하여 그 진동차가 적은 것을 볼 수 있다. 이와 같이 본 논문에서 제안된 개선된 RED 알고리즘을 적용하여 그 성능이 향상되는 것을 모의 실험 결과를 통하여 검증하였다.

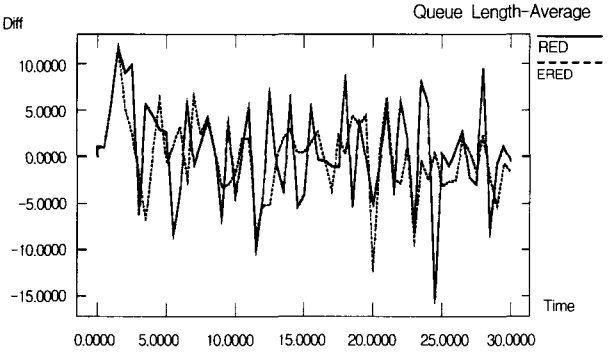
(그림 3)과 (그림 4)에서의 모의실험은 노드 수 각 16개, 전송속도 10MB, max_p는 0.05로 세팅하였다. (그림 3)에는 기존의 RED 알고리즘으로 모의실험을 한 결과 그래프가 나타나있고, (그림 4)에는 개선된 RED 알고리즘으로 모의실험을 한 결과 그래프가 나타나있다. (그림 3)과 (그림 4)에서의 모의실험 결과의 차이점을 알아보고자 (그림 5)에서는 RED와 ERED 각각의 실제 큐 길이에서 평균 큐 길이를 뺀 값을 도식화하였다.



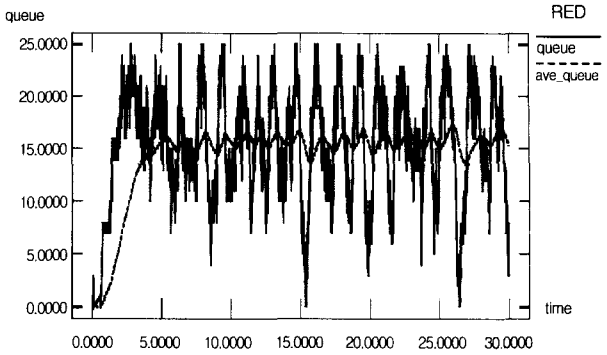
(그림 3) RED를 사용한 모의실험 결과(10MB)



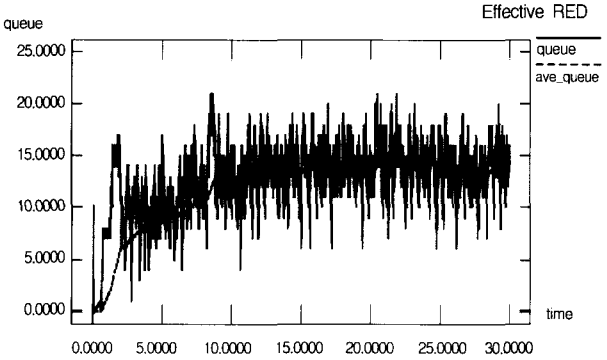
(그림 4) 개선된 RED를 사용한 모의실험 결과(10MB)



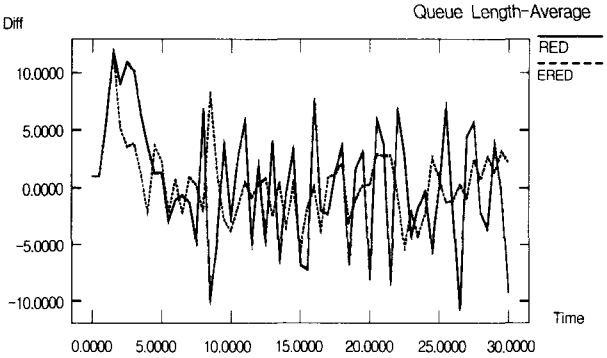
(그림 5) 실제 큐 길이와 평균 큐 길이의 차이(10MB)



(그림 6) RED를 사용한 모의실험 결과(100MB)



(그림 7) 개선된 RED를 사용한 모의실험 결과(100MB)

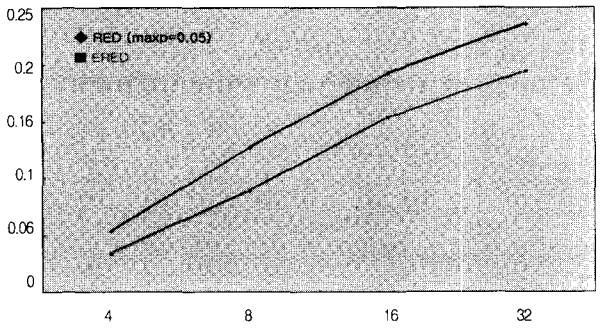


(그림 8) 실제 큐 길이와 평균 큐 길이의 차이(100MB)

(그림 6)과 (그림 7)에서의 모의실험은 나머지 환경은 동일하고, 전송속도만 100MB로 세팅하였다. (그림 6)에는 기존의 RED 알고리즘으로 모의실험을 한 결과 그래프가 나타나있고, (그림 7)에는 개선된 RED 알고리즘으로 모의실험을 한 결과 그래프가 나타나있다. (그림 8)과 (그림 9)에서의 모의실험 결과의 차이점을 알아보고자 (그림 8)에서는 RED와 ERED 각각의 실제 큐 길이에서 평균 큐 길이를 뺀 값을 도식화하였다.

(그림 8)에 나타난 실제 큐 길이의 변화를 살펴보면, RED 알고리즘에서는 트래픽이 버스트하여 실제 큐 길이의 진동차가 (그림 3)보다 크게 나타나며, 개선된 RED 알고리즘에서는 작게 나타나며 보다 빠르게 평형 상태로 접근하는 것을 볼 수 있다. 이것은 트래픽이 버스트하여 개선된 RED

알고리즘에서는 현재 트래픽을 빠르게 반영하여 평형상태로 가게 되는 것이다. 또한, (그림 7)에서와 같이 개선된 RED 알고리즘에서 실제 큐 길이의 진동차가 (그림 8)에서의 기존의 RED 알고리즘보다 현저하게 작은 것을 볼 수 있으며, 이것은 큐잉 지연시간이 현저하게 감소되는 것을 볼 수 있다.



(그림 9) RED와 ERED에서의 패킷 폐기율

RED 알고리즘을 사용하는 목적은 처리율을 높이고, 지연시간을 줄이는 것이다. (그림 9)는 RED와 ERED 알고리즘을 사용한 경우 패킷 폐기율을 나타낸 그래프이다. 그림에서도 알 수 있듯이 기존의 RED 알고리즘보다 개선된 RED 알고리즘에서의 패킷 폐기율이 감소한 것을 볼 수 있다.

본 논문에서 제시한 개선된 RED 알고리즘을 이용하여, 기존의 RED 알고리즘을 이용하는 것보다 처리율을 높였을 뿐 아니라, 지연시간이 줄어들게 됨을 모의실험을 통하여 검증하였다.

5. 결론 및 향후 과제

최근 네트워의 규모는 점점 확장되고 있으며, 다양한 시스템이 네트워에 연결되어 있으므로 인터넷은 끊임없는 변화에 직면하고 있으며, 트래픽은 가히 기하급수적으로 증가하고 있다고 볼 수 있다. 이러한 트래픽의 증가로 인하여 네트워에는 심각한 혼잡 상황이 발생하게 된다.

이와 같은 문제를 해결하기 위하여 RED(Random Early Detection) 알고리즘이 제시되었으며, RED 알고리즘은 기존의 문제점으로 지적되었던 전역 동기화 방지 및 패킷 손실률 감소에 성공적이었다. 반면에 RED 알고리즘이 트래픽 상태가 유동적으로 변경되는 네트워에 적용할 경우 발생되는 문제점이 지적되었고, 이것을 개선하기 위한 방법으로서 고정된 값을 가지는 패킷 폐기 확률을 적용하는 경우 현재 네트워 상황이 고려되지 않아 효율적으로 동작하지 못하는 점이다.

본 논문에서는 RED 알고리즘을 사용하는데 있어 매개변수 값을 현재 네트워 상황을 적절하게 반영할 수 있도록 매개변수 값을 동적으로 변경하는 방법을 제안하였다. 이를 위해 먼저 RED 매개변수 값이 게이트웨이의 큐에 미치는 영향을 분석하고, 잘못 설정된 매개변수 값을 사용한 경우 RED 알고리즘이 네트워 성능을 저하시킬 수 있음을 보여주었다. 그리고, 이와 같은 분석을 통하여 현재 트래픽의 상황에 따라 현재 큐 길이를 바탕으로 적절하게 매개변수 값을 동적으로 조정하였고, 또한 현재 큐 길이와 평균 큐 길이 및 이전 큐 길이를 비교하여 어느 정도의 가중치를 주는 부분을 두어 보다 정확하게 네트워 상태를 반영할 수 있도록 하였다. 네트워 트래픽의 변화에 따라 동적으로 최적의 매개변수 값으로 세팅되는 개선된 RED(Effective RED) 알고리즘을 제안하여 기존의 RED 알고리즘보다 안정적으로 동작하며, 평형상태(Equilibrium point)로 수렴함을 보여주었다.

앞으로의 연구 과제로는 좀더 다양한 토폴로지에 대한 검증과 여러 가지 상황을 변경하여 문제를 검증하는 것과 제안된 방법이 실제 네트워에서 유연하게 동작할 수 있도록 하기 위하여 좀 더 다양한 경우에 대해 고려한 연구와 다양한 환경에서의 실험 및 검증이 필요하다.

참 고 문 헌

[1] C. Lefelhocz, "Congestion Cortol for Best-Effort Service : Why We Need a New Paradigm," IEEE Network, January, 1996.

[2] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, January, 1997.

[3] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control," RFC 2581, April, 1999.

[4] Braden, B., "Recommendations on Queue Management and Congestion Avoidance in the Internet," RFC 2309, April, 1998.

[5] Jacobson, V., "Congestion Avoidance and Control," Proceeding of SIGCOMM88, August, 1988.

[6] Floyd, S., Fall, K., "Router Mechanisms to Support End-to-End Congestion Control," LBL Technical report, February, 1997.

[7] Floyd, S., Jacobson, V., "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transaction on Networking, August, 1993.

[8] Christiansen, M., Jeffay, K., Ott, D., and Smith F. D., "Tuning RED for Web Traffic," IEEE/ACM Transactions, June, 2001.

[9] Floyd, S., "Ns Simulator Tests for Random Early Detection(RED) Gateways," Technical report, October, 1996.

정 규 정

e-mail : kjjung@cs.kw.ac.kr
1999년 광운대학교 전자계산학과(학사)
2002년 광운대학교 컴퓨터과학과(공학석사)
2002년~현재 한화 S&C 컨설팅사업부
관심분야 : 컴퓨터 통신, 혼잡제어, QoS

이 동 호

e-mail : dhlee@kw.ac.kr
1979년 서울대학교 전자공학과(학사)
1983년 서울대학교 컴퓨터공학과(공학석사)
1988년 서울대학교 컴퓨터공학과(공학박사)
1997년~2001년 개방형컴퓨터통신연구회 Internet-KIG 부위원장, 위원장
2002년~현재 개방형컴퓨터통신연구회 TA부위원장
1984년~현재 광운대학교 컴퓨터과학과 교수
관심분야 : 프로토콜 성능평가, 혼잡제어, 인터넷 표준