

# TCP/IP 기반의 정지 위성 궤도 통신망에서 TARED 알고리즘 성능 분석

준회원 서진원, 정회원 김덕년

## Performance Analysis of Target Adapted RED Algorithm on TCP/IP based GEO Satellite Communication Network

Jin-won Seo Associate Member, Doug-nyun Kim Regular Members

### 요 약

위성망과 같이 DBP(Delay Bandwidth Product)가 큰 고속망에서 전송 지구국의 버퍼는 일시적인 막힘 현상에 의한 전송 효율이 저하되는 것을 방지하기 위하여 큐 현상을 충분히 소화하도록 설계되어야 한다. 뿐만 아니라 버퍼의 어떠한 상태 하에서도 막힘 없는 유연한 전송 패킷의 흐름과 수많은 연결 트래픽 간의 공정성을 보장하고 접속 서비스의 특성에 따라서 정의 되는 QOS 를 충족시키는 기술이 중요하다. 기존의 버퍼알고리즘들은 여러 네트워크 환경의 다양한 변화에 적절히 대응하지 못하여 위성망과 같은 특수한 네트워크에 적합하지 않다. Floyd에 의해 제안된 RED(Random Early Detection) 알고리즘은 기존의 버퍼 알고리즘과 비교하여 뛰어난 버퍼관리 성능을 보여왔다. 하지만 RED 알고리즘 방식은 TCP 패킷의 연결 수나 네트워크의 변화에 적절히 대응하지 못하여 위성망과 같이 다수의 사용자가 동시 다발적으로 발생할 수 있는 환경에서는 적합하지 못하다. 본 논문에서는 입력 비율에 따라 향상된 버퍼 활용도를 가진 TARED(Target Adaptive RED) 알고리즘이 위성망을 포함한 네트워크 환경에서 향상된 성능을 보이고 제안된 TARED 알고리즘을 이용하면 입력 비율에 따른 버퍼의 점유 레벨의 오버플로우나 언더플로우가 없이 전체 시스템의 성능을 향상 시킴과 동시에 네트워크의 적응능력과 시스템의 안정화에 우수함을 보일 수 있었다.

Key Words : ARED, TARED, TCP/IP, Satellite communication

### ABSTRACT

We must design the buffer algorithm that protects traffic congestion and decreasing throughput at satellite communication network. It is important that buffer algorithm is satisfied with the good performance of transmission packet, responsibility of many connecting traffic and the QOS for connecting character. Old buffer algorithms are not the suitable algorithms when we have the satellite communication network environment. RED buffer algorithm is proposed by Floyd. It has a better performance than old buffer algorithm. But this algorithm is not well adapted a number of connecting TCP packet and changing network, so this algorithm has a bad performance on satellite communication network that is many of connecting user at same time. This paper propose the TARED(Target Adaptive RED). It has a good performance, adaptation and stability on satellite communication network and has not overflow and underflow of the buffer level.

### 1. 서론

지난 몇 년간 인터넷의 발전에 따라 다양한 어플리케이션을 수용하기 위해 TCP/IP 네트워크를 위한

\*명지대학교 통신공학과  
논문번호 : 030560-1216, 접수일자 : 2003년 12월 16일

큐 알고리즘이 발전되어 왔다. 하지만 위성망과 같이 DBP 와 패킷 지연 그리고 에러율이 큰 고속망에서는 추가적인 버퍼관리가 필요하다. 버퍼 관리에서 트래픽의 원활한 흐름을 보장하기 위하여 입력되는 패킷으로 인한 버퍼의 오버 플로우가 발생하지 않도록 적절히 패킷을 폐기시키며 각 연결의 상태 관리와 신규 및 기존의 패킷에 대한 관리가 필요하다[1]. 그와 동시에 전송률을 높이고 전송지연을 최소화하는 일은 간단한 문제가 아니다. 기존의 Drop tail 방식의 버퍼 알고리즘은 버퍼 내의 공간이 다 채워질 때까지 들어오는 모든 패킷을 수용하는 알고리즘이다. 하지만 Drop tail 버퍼 알고리즘의 단점은 버퍼의 공간이 다 채워진 후에는 들어오는 모든 패킷들이 폐기되어 네트워크상의 혼잡을 가중시키고 사용자가 서비스를 이용하는데 높은 지연을 유발한다는 것이다. 이러한 단점을 극복하기 위하여 AQM(Active Queue Management)이 발달하였으며, RED(Random Early Detection)알고리즘[2] 이 Floyd에 의해 제안되었다. RED의 특징은 기존의 Drop tail 과는 달리 입력되는 패킷을 미리 확률적으로 폐기하는 것이다. 따라서 일정한 평균 큐 길이를 유지하고 빠르게 네트워크의 혼잡을 감지한다. 즉 버퍼 내에서 평균 큐 길이를 계산하여 정의된 일정 구간 안에 큐 길이가 포함되면 들어오는 패킷에 대한 패킷 폐기 확률을 계산하여 미리 패킷을 폐기하는 방식이다. 패킷을 미리 폐기하여 Drop tail 알고리즘의 단점인 많은 수의 패킷 폐기를 사전에 예방할 수 있으며 큐 지연과 네트워크의 혼잡을 예방할 수 있다. 그렇지만 RED 알고리즘의 성능은 네트워크의 혼잡 정도와 RED 알고리즘의 파라미터 그리고 실질적인 사용자 연결 수에 민감하게 반응한다. 만일 링크가 빠르게 혼잡해지거나 RED 파라미터의 최대 패킷 폐기 확률 값이 높을 때는 평균 큐 길이가 작아지게 되고 반대로 링크의 트래픽이 적거나 최대 패킷 폐기 확률 값이 낮을 때는 평균 큐 길이가 커지게 된다. 이것은 RED 알고리즘 목적 중에 하나인 평균 큐 길이를 일정하게 유지하는 것을 어렵게 만들고 네트워크의 효율을 떨어뜨린다. 또한 큐 지연은 인터넷 사용자가 서비스를 이용하는데 있어서 중요한 요소가 된다. 따라서 네트워크 설계자는 우선적으로 평균 큐 길이를 추정할 수 있어야 하며 RED 알고리즘의 파라미터를 현재의 트래픽 환경에 맞게 조절할 수 있어야 한다. RED 알고리즘의 단점은 전송률이 트래픽의 상황과 RED 알고리즘 파라미터에 따라 민감하게 반응한다는 것이다. 부분적으로 RED 알고리즘은 평균 큐 길

이가 버퍼의 최대 문턱값보다 크면 입력되는 모든 패킷을 폐기하여 전송률의 감소와 패킷 폐기확률의 증가로 이어진다. 따라서 RED 알고리즘 파라미터 값을 네트워크의 환경에 알맞게 변화시켜 주어야 한다. ARED 알고리즘[3][4][5] 은 RED 알고리즘의 기본 구조를 유지하면서 어떠한 환경에서도 버퍼내의 목표치 사이에 평균 큐 길이를 일정하게 유지하기 위해 최대 폐기 확률 값을 조절하는 알고리즘이다. ARED 알고리즘은 RED 알고리즘의 다른 이점을 희생하지 않고 다양한 환경에서의 평균 큐 길이를 성취한다. 이것은 평균 큐 지연의 예측뿐만 아니라 버퍼 최대값으로의 접근 가능성도 낮출 것이다. ARED 알고리즘은 패킷의 손실 비율과 큐 지연의 다양성을 감소시키며 기존의 RED 알고리즘의 장점을 버리지 않고 평균 큐 길이를 안정화하고 RED 알고리즘 파라미터를 향상시킨다. 하지만 RED 와 ARED 알고리즘들은 평균 큐 길이를 안정화하는 것에만 목적을 두어 현재 네트워크의 상황에 따른 버퍼의 활용도에 대해서는 고려하지 않았다. 본 논문에서 제안하는 TARED(Target Adapted RED)는 매 샘플마다 버퍼로 입력되는 입력 비율을 계산하여 네트워크의 혼잡정도를 판단해서 버퍼의 활용도와 전송률을 향상시킬 수 있다. 2장에서는 RED 와 ARED 알고리즘에 특성에 대해 알아보고 새롭게 제안된 TARED 에 대해 자세히 알아보도록 한다. 3장에서는 제안된 알고리즘의 성능을 알아보기 위해 시뮬레이션 환경을 구성하고 모의실험을 통해 성능을 평가한다. 마지막 4장에서는 제안된 알고리즘의 결과를 토대로 결론 및 향후 연구 방향에 대해 알아보도록 한다.

## II. 기존의 AQM 알고리즘과 TARED 알고리즘

이 장에서는 기존의 AQM 알고리즘 중에 RED 와 ARED 알고리즘에 대해 알아본 후 ARED 알고리즘을 향상시킨 TARED에 대해 알아보도록 한다.

### 1. RED 알고리즘

Floyd에 의해 제안된 RED 알고리즘은 버퍼 관리에서 네트워크의 원활한 흐름을 보장하기 위하여 입력되는 패킷 때문에 오버플로우가 발생하지 않도록 적절히 입력된 패킷을 폐기하는 알고리즘이다. 그림 1 은 RED 알고리즘 순서도이다.

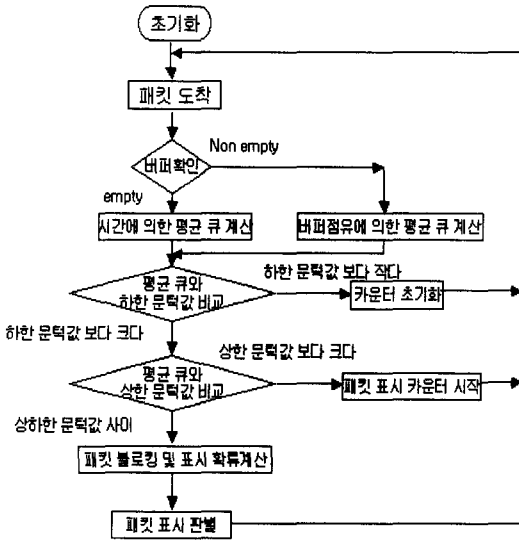


그림 1. RED 알고리즘

RED 알고리즘은 각 패킷이 라우터에 도착할 때마다 평균 큐 길이를 계산하여 평균 큐 길이가 하한 값보다 작으면 패킷을 폐기하지 않고 상한 문턱값보다 평균 큐 길이가 크면 모든 패킷을 폐기한다. 평균 큐 길이가 하한 문턱값과 상한 문턱값 사이에 있을 때는 패킷 폐기확률을 계산하여 패킷을 폐기한다. 즉 패킷이 폐기되는 것은 평균 큐 길이에 의지한다. 평균 큐 길이의 계산은 식 (1)과 같으며 과도한 트래픽의 영향을 적게 받기 위해 Wq(weight queue)를 사용하여 적절히 대응한다.[2]

<버퍼가 비어있는 경우>

$$m = \text{버퍼가 비어있는 시간} \times \text{입력전송률(packets)}$$

$$Avg = \{(1 - Wq)m\} \times Avg \quad (1)$$

<버퍼가 비어있지 않는 경우>

$$Avg = (1 - Wq) \times Avg + Wq \times q$$

Avg : 평균큐

Wq(weight queue) : 가중치

q : 현재 점유된 큐의 크기

m : 패킷수

평균 큐 길이는 실질적인 연결수 N 이 증가할수록 상한 문턱값에 도달하여 모든 패킷이 폐기된다. 따라서 제한된 버퍼를 갖는 라우터는 작은 N 조차

쉽게 상한 문턱값을 초과할 수 있다. 입력된 패킷의 블로킹 및 패킷 표시 확률은 식 (2) 와 같으며 최대 블러킹 확률은 평균 큐 크기가 상한 문턱값에 도달 하였을 때 블러킹 확률 Pb의 최대값이며 Pb는 평균 큐가 상하한 문턱값 사이에 있을 때 0에서 최대 블러킹 확률 사이에서 선형적으로 변하게 된다.

$$p_b = \frac{\max p (Avg - \min th)}{\max th - \min th}$$

$$p_a = \frac{p_b}{1 - count \times p_b} \quad (2)$$

$p_b$  : 패킷 블러킹 확률

$p_a$  : 패킷 표시 확률

max p : 최대 블러킹 확률

maxth : RED 버퍼의 상한 문턱값

minth : RED 버퍼의 하한 문턱값

count : 이전 표시된 패킷 이후 표시 안된 패킷의 수

이러한 제한된 RED 알고리즘의 성능을 보완하기 위한 여러 가지 알고리즘이 제안되었다.

## 2. ARED 알고리즘

RED 알고리즘의 목적은 일정한 평균 큐 길이의 유지와 빠른 혼잡을 감지하는 것인데 이것을 통해 낮은 평균 큐 지연과 높은 전송률에 도달하는 것이다. 그러나 RED 알고리즘은 평균 큐 길이가 파라미터의 설정값과 네트워크의 혼잡 정도에 따라 민감하게 변한다는 것이다. 지연은 사용자가 서비스를 사용 하는데 중요한 요소가 된다. 따라서 RED 알고리즘 파라미터를 현재의 트래픽 환경에 맞게 조절해주어야 한다. 또 다른 RED 알고리즘의 취약점은 전송률 역시 트래픽의 상황과 RED 알고리즘 파라미터에 민감하다는 것인데 종종 RED 알고리즘의 평균 큐 길이가 상한 문턱값보다 크게 되었을 때 RED 알고리즘 자체가 잘 수행되지 못해 결과적으로 전송률의 감소와 패킷 폐기확률의 증가로 이어진다. 이러한 이유로 RED 알고리즘 파라미터 값을 변화시켜줄 필요가 있다. 그림 2는 ARED 알고리즘 순서도이다.

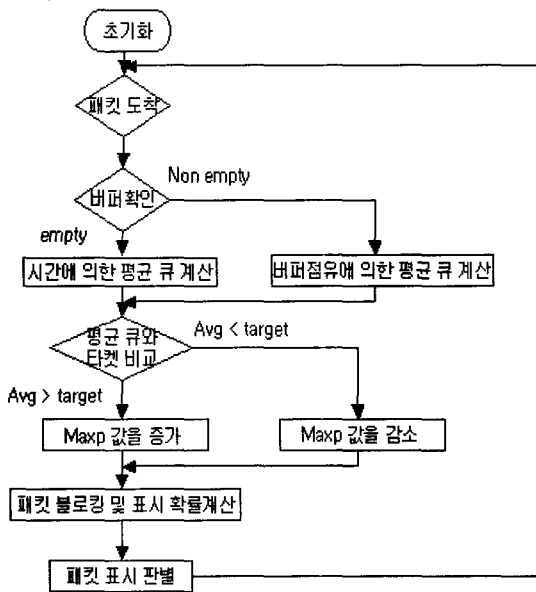


그림 2. ARED 알고리즘

ARED 알고리즘의 목적은 RED 알고리즘의 기본 구조를 유지하면서 버퍼의 큐 길이 목표치 사이에 평균 큐 길이를 유지하기 위해 최대 블러킹 확률값을 실시간으로 조절하는 것이다. ARED 알고리즘은 RED 알고리즘의 다른 이득을 희생하지 않고 다양한 환경, 특히 위성망을 포함한 여러 네트워크에서 일정한 평균 큐 길이를 성취한다는 것이다. 이것은 평균 큐 길이의 예측뿐만 아니라 평균 큐가 오버플로우 되는 가능성을 낮출 것이다. 그러므로 ARED 알고리즘은 패킷 손실 비율과 평균 큐 길이를 예측할 수 있다. ARED 알고리즘은 평균 큐 길이를 안정화하고 RED 알고리즘의 파라미터를 자동으로 향상시킨다. RED 알고리즘을 기본으로 변화된 ARED 알고리즘의 대해 살펴보면 식 (3)과 같다. ARED 알고리즘의 동작을 살펴보면 RED 알고리즘과 같이 평균 큐 길이를 구한 후 그 길이를 목표치와 비교하여 평균 큐 길이가 목표치 보다 높으면 최대 블러킹 확률값을 증가시켜 패킷 폐기 확률을 높이고 평균 큐 길이가 목표치 보다 낮으면 최대 블러킹 확률값을 감소시켜 패킷 폐기확률을 떨어뜨린다. ARED 알고리즘의 다른 접근법을 알아보자. 최대 블러킹 확률값은 단지 평균 큐 길이를 목표치 사이에 유지시키는 것이 아니라 여러 트래픽의 상황에 맞게 패킷 폐기 확률을 변화시켜 평균 큐 길이를 적응시키는 것이다. 즉 일정한 목표점에 평균 큐 길이를 유지시키는 것

이다. 또한 전형적인 round trip time 보다 큰 시간으로 천천히 적응시킨다. 최대 블러킹 확률값은 0.01에서 0.5 범위에서 유지되도록 강요되는데 이것은 평균 큐 길이의 변화에 대응하기 위해 패킷 폐기 확률을 천천히 변화해야 하기 때문이다.

매 시간 마다 :

If ( avg > target and maxp <= 0.5)  
 maxp 증가 ;  
 maxp ← maxp \* α ;  
 Else if ( avg < target and maxp >= 0.001)  
 maxp 감소 ;  
 Maxp ← maxp \* β ;

Variables :

Avg : 평균 큐 길이 (3)

Fixed parameters :

Interval : time ; 0.5 seconds

Target : 평균 큐 길이를 위한 목표치 ;

[minth + 0.4 \* (maxth - minth),

minth + 0.6 \* (maxth-minth)]

: incremnet ; min (0.01, max/4) :

decrease factor ; 0.9

우리는 [0.01, 0.5] 사이에 최대 블러킹 확률값을 제한하므로 성능의 과다한 저하를 예방할 수 있다. 우리는 최적의 ARED 알고리즘에 대해 확정하지 않았지만 ARED 알고리즘은 기존의 RED 알고리즘과 비교하여 보다 광범위하게 사용될 것으로 보이고 인터넷에서 지금의 RED 알고리즘 보다 안전하게 작동할 것이다. 최대 블러킹 확률값의 적절한 적응의 결과 ARED 알고리즘은 위성망과 같은 다수의 접속자와 높은 지연을 유발하는 네트워크 환경에서 좋은 성능을 나타낼 것이다.

### 3. TARED 알고리즘

지금까지 소개된 AQM 알고리즘의 목적은 모두 평균 큐 길이를 버퍼 내에서 일정하게 유지하여 많은 수의 패킷이 폐기되는 것을 방지하여 평균 큐 길이를 일정하게 유지시키고 전송률을 향상시키는 것이었다. 그렇지만 버퍼의 효율 면으로는 기존의 AQM 알고리즘들은 높은 성능을 보이지 못하였다. 일정한 큐 길이 유지하는 시스템의 안전성을 보장하였지만 전체 전송률의 성능은 크게 고려되지 못했다. 본 논문에서 제안하는 TARED (Target Adapted

Random Early Detection) 알고리즘은 라우터의 패킷 입력비율을 계산하여 그에 따라 버퍼의 목표치를 조절하여 버퍼의 활용도를 증가시키며 시스템의 안정성도 보장된다.

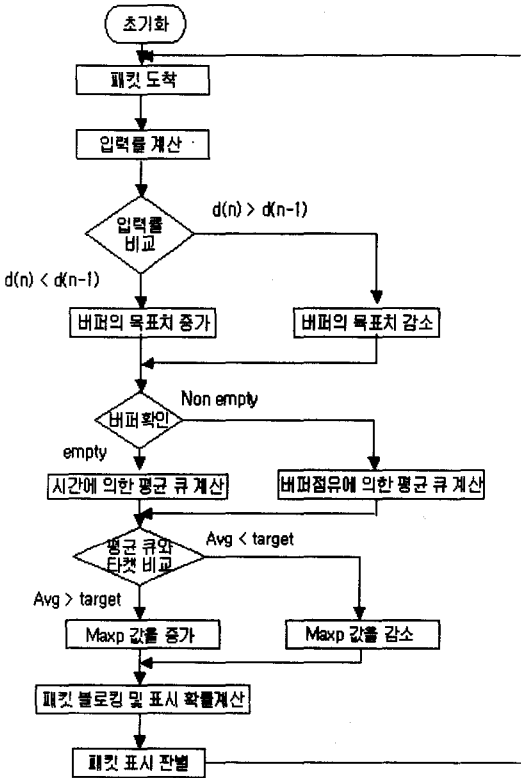


그림 3. TARED Algorithm

그림 3은 TARED 알고리즘의 순서도이다. TARED 알고리즘은 샘플 간격마다 라우터로 들어오는 패킷의 입력률을 계산하여 현재의 트래픽이 안정적이면 버퍼의 목표치를 높여서 전송률을 증가시키고 트래픽이 과다하거나 불안정적이면 버퍼의 목표치를 낮게 조절하여 시스템의 안정성을 보장한다.

식 (4) 와 같이 매 샘플마다 패킷 입력률을 구한 후 입력 가중치 k 를 이용하여 평균 입력률을 구한다. 현재 평균 입력률과 바로 전 평균 입력률을 비교하여 현재 평균 입력률이 작을 때는 트래픽이 안정적이라고 판단하여 버퍼의 목표치를 증가시키고 반대로 현재 평균 입력률이 전 평균 입력률보다 크게 되면 트래픽이 증가하거나 불안정하다고 판단하여 버퍼의 목표치를 떨어뜨리게 된다.

매 샘플 간격마다 :

$$D(n) = q(n) - q(n-1)$$

$$d(n) = (1-k)d(n-1) + kD(n)$$

If  $(d(n) < d(n-1) \text{ and } T(n) \leq \text{maxth } T)$

$$T(n) \text{ 증가 ;}$$

$$T(n) \leftarrow T(n) * X ;$$

Else if  $(d(n) > d(n-1) \text{ and } T(n) \geq \text{minth } T)$

$$T(n) \text{ 감소 ;}$$

$$T(n) \leftarrow T(n) * Y ; \tag{4}$$

parameters :

Interval : time ; 0.5 seconds

$q(n)$  : 큐 길이

$D(n)$  : 입력비율

$d(n)$  : 평균 입력 비율

k : 입력가중치

$T(n)$  : 평균 큐 길이를 위한 목표치(B/2)

minth T : 최저 목표치

maxth T : 최고 목표치

increase factor X ; 1.1

decrease factor Y ; 0.9

### 3.1 파라미터 설정

평균 입력 비율을 구할 때 파라미터 k 는 입력 비율의 반응 속도를 나타낸다. 이 값을 너무 작게 하였을 경우에는 패킷 입력률 변화에 목표치가 둔감하게 변화하게 되어 현재의 트래픽 상황을 고려하지 못하여 과다한 트래픽 발생시 미리 예상하지 못하고 버퍼의 오버플로우가 날 수 있다. 반면 너무 크게 하였을 경우에는 목표치가 빠르게 진동하여 낮은 전송률을 나타내게 된다. 시뮬레이션 결과 0.002 값으로 설정하면 최적의 성능을 발휘한다. 최저 목표치의 경우 시스템의 최저 전송률과 네트워크의 상황에 따라 설정하게 되는데 최저 큐 길이를 버퍼의 중간 지점이하로 내려가는 것을 방지하기 위해 버퍼의 중간 지점으로 설정한다.

$$\sigma = \sqrt{\frac{100}{n} \sum_{n=0}^{100} (x_n - T)^2 / 100} \tag{5}$$

$x_n$  : 샘플 큐 길이

T : 버퍼의 목표치

최고 목표치의 경우 버퍼의 오버플로우를 예방하면서 버퍼의 활용도는 최대한 증가시키는 범위를 설정하는 것이 중요한데 큐 길이가 목표치에 대한 표준편차와 트래픽 상황에 맞추어 조절할 수 있다. 최대 목표치에 대한 큐 길이의 표준 편차값이 크게 되면 큐 길이의 진폭 변화폭이 크게 되어 버퍼의 오버플로우나 시스템의 불안정함을 야기하므로 안정된 최대 목표치를 설정하여야 한다. 또한 급격히 트래픽이 증가할 때 버퍼의 오버플로우가 날 수 있다. 본 논문의 시뮬레이션 결과 버퍼의 70 %의 최대 목표치를 설정하였을 경우 최적의 성능을 발휘하였다.

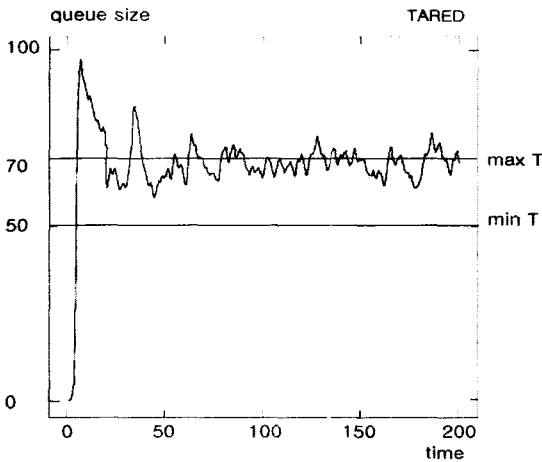


그림 4. TARED 파라미터 X = 1.1

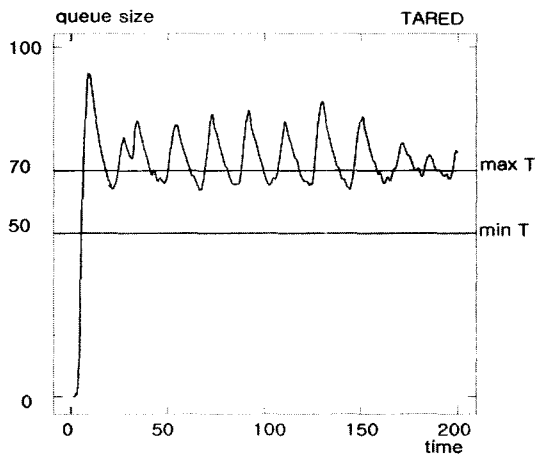


그림 5. TARED 파라미터 X = 1.2

목표치 증가 요소인 X 값은 평균 큐 길이가 최대 목표치에 도달하는 시간과 최대 목표치에 대한 큐 길이의 표준편차 값과 관련이 있다. 그림 4와 같이 X 값을 1.1 값으로 설정하였을 경우 빠른 샘플 시간 안에 최대 목표치에 도달하게 되며 최대 목표치에 대한 표준편차 값 역시 최저값  $\sigma = 1.73$  을 갖는다. 그림 5와 같이 그 이상의 값으로 설정하였을 경우에는 큐 길이가 최대 목표치 근처에 도달하기 어려워지며 최대 목표치에 대한 큐 길이의 표준편차 역시 큰 값  $\sigma = 3.25$  으로 나와 큐 길이가 불안정해지고 버퍼의 오버플로우가 생기게 된다. 목표치 감소 요소인 Y 값 역시 0.9 값으로 설정하였을 때 최저 목표치에 대한 큐 길이의 표준편차가 최저값을 가지게 되므로 최고의 성능을 나타내게 된다.

### III. 모의실험 및 성능분석

ARED 알고리즘과 TARED 알고리즘의 성능을 확인하기 위해 위성망을 포함한 네트워크 환경을 그림 6 와 같이 구축하였다. 네트워크는 위성망을 포함한 것으로 100 패킷의 버퍼용량을 가진 라우터를 경유하여 200초 동안 10개의 소스에서 TCP/FTP 데이터를 전송하는 NS - 2 시뮬레이션이다. 지상 링크의 대역폭은 45Mbps로 설정하며 전송지연은 10ms로 하였다.

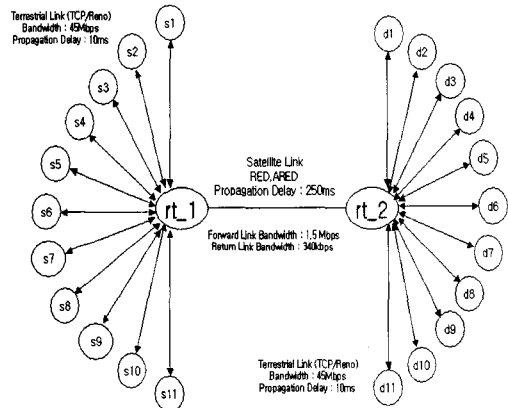


그림 6. NS-2 네트워크 구성도

위성링크의 전송지연은 GEO 위성을 가정하여 250ms로 설정하였으며 위성채널의 순방향 대역폭과 역 방향 대역폭은 각각 1.5Mbps, 340Kbps로 비대칭 링크를 구성하였다. 또한 지상망 보다 높은 에러률을

위성링크에 적용시켜 에러를 마다의 전송률을 비교 하였다. 표 1 과 같이 ARED 와 TARED 알고리즘 들의 파라미터들을 설정한 후에 그림 6 의 시뮬레이션 환경에 차례로 적용을 시켜 보았다.

표 3. 시뮬레이션 파라미터

Parameter		Value	
Network	Application Layer	FTP	
	Transmission Layer	TCP Reno	
	Simulation Time	200 sec	
	Terrestrial Bandwidth	45 Mbps	
	Terrestrial Propagation Delay	10 ms	
	Satellite Link Bandwidth	Forward	1.5Mbps
		Return	340kbps
	Satellite Link Propagation Delay	250 ms	
	Satellite Link Packet Error	0.0005 0.005 0.001 0.05 0.01	
ARED	Maximum threshold (maxth)	20	
	Maximum threshold (maxth)	70	
	Queue weight (wq)	0.002	
	Buffer size (B)	100	
	Target range (T)	[40.50]	
TARED	Minimum Target (min T)	50	
	Maximum Target (max T)	70	
	Queue weight (wq)	0.002	
	Buffer size (B)	100	
	Input rate weight (k)	0.002	
	Target (T)	50	

그림 7, 8 은 안정적인 트래픽 상황에서 ARED 알고리즘의 큐 길이와 TARED 알고리즘의 큐 길이에 대한 것이다. 그림 7 의 ARED 알고리즘의 큐 길이는 목표치 근처에서 큐 길이가 일정하게 유지되는 것을 알 수 있다. 이것은 기존의 RED 알고리즘과 비교하여 안정된 큐 길이를 유지하여 시스템의

안정성은 유지하지만 버퍼의 활용이나 전송률에서는 만족스러운 성능을 내지 못하는 것을 알 수 있다. 반면 그림 8 의 TARED 알고리즘의 경우 ARED 알고리즘과는 달리 목표치를 실시간으로 적용시켜 큐 길이를 최대 목표치까지 활용하여 버퍼의 70 % 까지 사용하는 것을 알 수 있다.

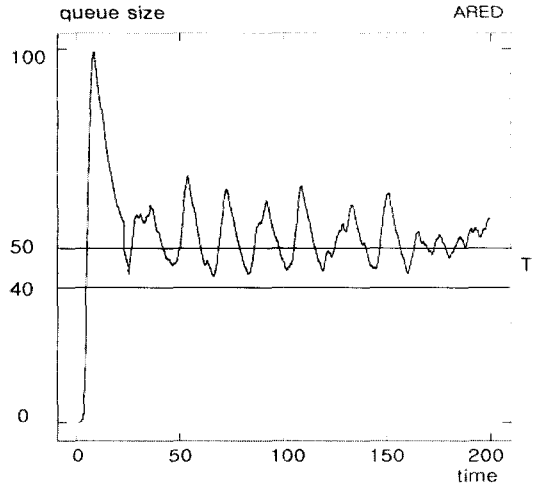


그림 7. ARED 큐 길이

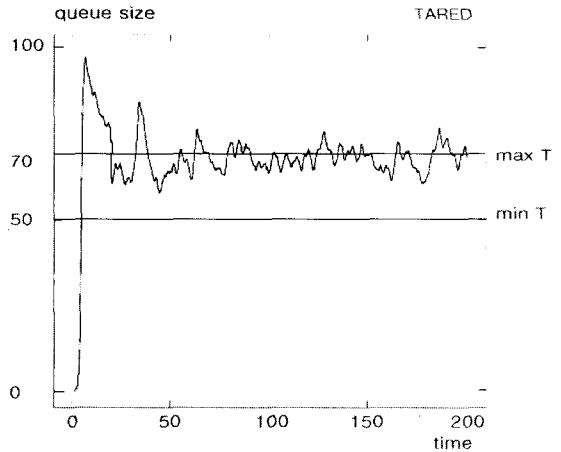


그림 8 TARED 큐 길이

그림 9, 10 은 그림 7, 8 의 패킷 폐기 확률을 구한 것이다. 그림 9 의 ARED 알고리즘의 패킷 폐기 확률은 최대 블러킹 확률값이 0.01 에서 0.5 까지 실시간으로 변화하면서 심하게 변화하는 것을 알 수 있다. 그렇지만 그림 10 의 TARED 알고리즘의 패킷 폐기 확률은 목표치를 조절함에 따라 폐기 확률

이 안정화 되는 것을 알 수 있다. 이것은 시스템의 안정성을 높이고 전송률 또한 향상시키는 결과로 이어진다.

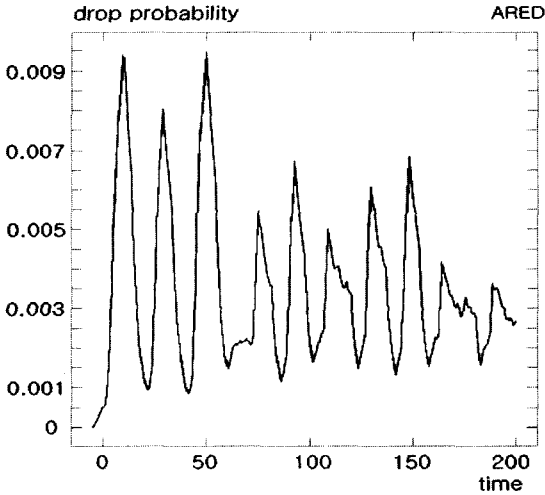


그림 9. ARED 패기 확률

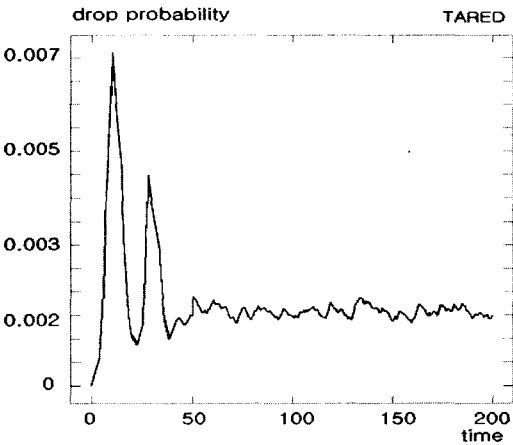


그림 10. TARED 패기확률

이렇듯 안정적인 트래픽 상황에서의 TARED 알고리즘은 기존의 ARED 알고리즘과 비교하여 버퍼의 활용도와 패킷 폐기 확률의 향상을 가져오는 것을 확인하였다. 그렇지만 여러 상황이 발생할 수 있는 네트워크를 고려한다면 갑자기 트래픽 혼잡이 가중되었을 때의 상황도 고려하여야 한다. 그림 11, 12은 네트워크에 일시적으로 트래픽이 증가하였을 경우에 대해 나타내었다. 그림 11의 ARED 알고리즘의 경우 70초 부분에서 트래픽이 일시적으로 증가하

여 큐 길이가 증가하였지만 최대 블러킹 확률값의 적응으로 인해 다시 목표치에 큐 길이가 돌아오는 것을 알 수 있다. 이것은 기존의 AQM 알고리즘들의 목적인 일정한 큐 길이를 유지하는 것에는 뛰어난 성능을 가지나 역시 버퍼의 활동이나 전송률에는 그림 12의 TARED 알고리즘 보다 떨어지는 것을 알 수 있다. TARED 알고리즘은 일시적인 트래픽 증가 시에 일시적으로 큐 길이가 증가하지만 목표치를 떨어뜨려 큐 길이를 감소시킨다. 하지만 빠른 시간 안에 목표치를 조절하여 다시 최대 목표치에서 큐 길이가 유지되는 것을 알 수 있다.

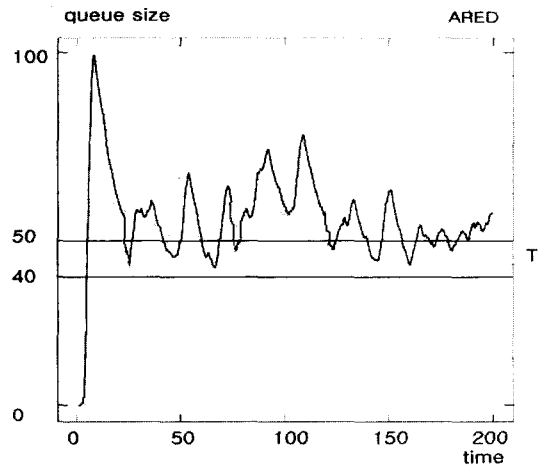


그림 11. 혼잡 트래픽의 ARED 큐 길이

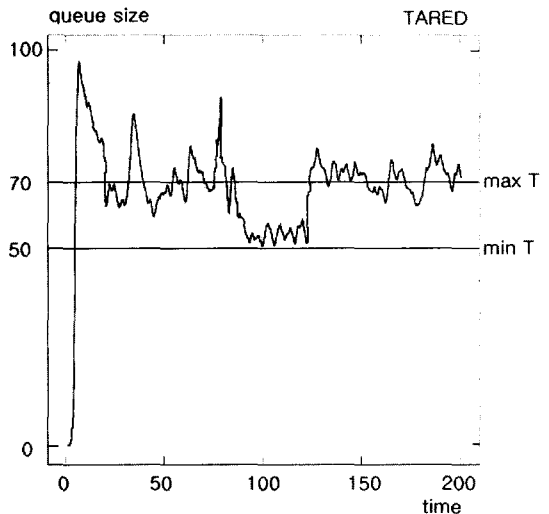


그림 12. 혼잡 트래픽의 TARED 큐 길이



그림 13, 14 는 그림 11,12 에 따른 패킷 폐기 확률이다. 그림 13 에서 알 수 있듯이 ARED 알고리즘의 경우 일시적으로 올라간 큐 길이를 목표치까지 떨어뜨리기 위해 일시적으로 폐기확률이 증가하는 것을 살펴볼 수 있다. 그림 14 의 TARED 알고리즘의 경우 일시적인 트래픽의 증가로 폐기 확률이 증가하지만 목표치의 조절로 패킷 폐기 확률이 안정화되는 것을 알 수 있다.

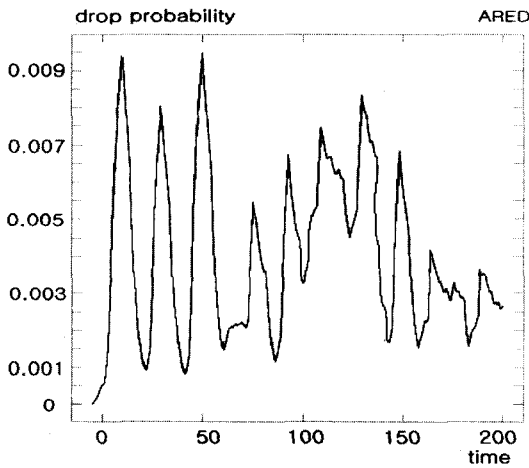


그림 13. 혼잡 트래픽의 ARED 폐기확률

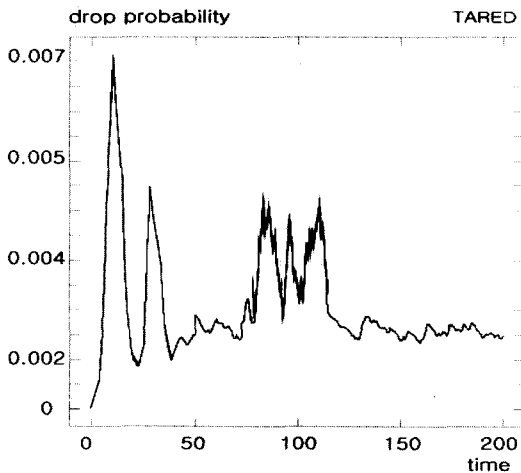


그림 14. 혼잡 트래픽의 TARED 폐기확률

이와 같이 TARED 알고리즘은 ARED 알고리즘과 비교하여 뛰어난 버퍼 활용 능력과 전송률을 가지고 있으며 네트워크에 과다한 트래픽이 발생하는

상황에서도 안정된 성능을 보이고 있다. 그림 15 는 위성링크의 특성인 큰 에러율에 따른 전송률을 나타낸 것이다. 트래픽이 혼잡하였을 경우와 패킷 에러 비율에 따라 모두 TARED 알고리즘이 ARED 알고리즘보다 향상된 전송률을 나타내고 있다.

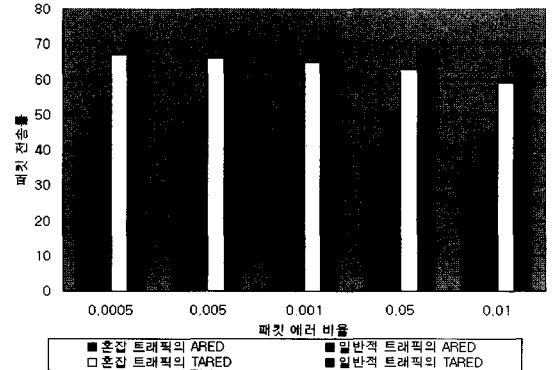


그림 15. 위성링크 에러율에 따른 전송률

이와 같이 인터넷과 그에 따른 서비스의 발전으로 고용량, 초고속 인터넷의 사용이 증가되는 시점에서 TARED 알고리즘은 기존의 AQM 알고리즘 보다 뛰어난 성능을 내고 있다.

#### IV. 결 론

RED 알고리즘의 파라미터들의 민감함은 여러 논문에서 논의되고 발전되어 왔다. ARED 알고리즘은 RED 알고리즘 파라미터 중 최대 블러킹 확률값과 상한 문턱값을 조절하여 평균 큐 길이를 보다 일정하게 유지시키는 알고리즘이다. 그렇지만 기존의 AQM 알고리즘들과 마찬가지로 평균 큐 길이를 일정하게 유지시켜 패킷 폐기확률을 줄이고 전송률을 높이는 것에 목적을 두어서 여러 네트워크의 상황에 맞는 목표치의 조절과 버퍼의 활용 등에 대해서는 고려하지 않았다. 하지만 본 논문이 제시하는 TARED 알고리즘은 실시간으로 버퍼에 입력되는 패킷 입력 비율을 계산하여 버퍼의 활용도의 증가와 전송률의 증가 그리고 시스템의 안정화까지 모두 고려하여 기존의 ARED 알고리즘 보다 파라미터에 덜 민감하고 버퍼의 활용과 전송률을 증가시키는 결과를 얻게 되었다. TARED 알고리즘은 초고속, 고용량의 데이터의 사용이 증가되는 현 네트워크 상황과 위성망과 같은 성질을 갖는 네트워크에서 보다 뛰어난 성능을 발휘할 것이다.

참고 문헌

[1] B. Braden et al., "Recommendation on Queue Management and Congestion Avoidance in the Internet." IETF RFC 2309, Apr.1998.

[2] S. Floyd, V. Jacobson, "Random Early Detection Gateway for Congestion Avoidance", ACM computer Communications Review, vol 24, Oct 1991.

[3] Sally Floyd, Ramakrishna Gummadi, and Scott Shenker "Adaptive RED: An algorithm for Increasing the Robustness of RED's Active Queue Management", AT&T Center for Internet Research at ICSI, August 1,2001, under submission.

[4] W.feng, D.Kandlur, D.Saha, and K.Shin. Techniques for Eliminating Packet Loss in Congested TCP/IP Network. U. Michigan CSE TR 349 97, November 1997.

[5] W.feng, D.Kandlur, D.Saha, and K.Shin. A Self Configuring RED Gateway. Infocom, Mar 1999.

[6] Victor Firoiu and Marty Borden. A Study of Active Queue Management for Congestion Control. Infocom, pages 1435 1444, 2000.

[7] V. Jacobson, K. Nichols, and K. Poduri, RED in a Different Light, September 1999. draft,www.cnafe.infn.it/~ferrari/papers/ispn/red\_light\_9\_30.pdf.

[8] M.May, J. Bolot, C. Diot, and B. Lyles. Reasons Not to Deploy RED. Proc. Of 7th. International Work shop on Quality of Service(IWQoS'99), pages 260 262, June 1999.

[9] T. Ziegler, S. Fdida, and C. Brandauer. Stability Criteria for RED with Bulk data TCP Traffic, 2001. Technical Report, August 1999.

서진원 (Jin-won Seo)

준회원



2002년 2월 명지대학교 졸업  
2004년 2월 명지대학교  
통신공학과 대학원 졸업

<관심분야> 무선통신, 위성통신

김덕년 (Doug Nyun Kim)

정회원



1971-1975년 서울대학교  
전기공학 학사  
1980-1981년 SUNY at Stony  
Brook 전기공학 석사  
1985-1988년 Auburn  
University 전기공학 박사  
1988-1995년 한국전자통신연구  
소(ETRI) 책임연구원

1995-현재 명지대학교 통신공학과 교수

<관심분야> 무선통신, 초고속 위성통신