

# 사설 망의 공정성을 향상시키기 위한 큐 관리 알고리즘

## (A Queue Management Algorithm for Improving Fairness in a Private Network)

강 태 형 <sup>†</sup> 구 자 현 <sup>†</sup> 정 광 수 <sup>\*\*</sup>

(Taehyung Kang) (Jahon Koo) (Kwangsue Chung)

**요 약** 인터넷이 급속하게 발달함에 따라 네트워크 사용자는 텍스트 기반의 서비스에서 멀티미디어 서비스 사용을 위한 고속의 네트워크를 요구하고 있다. 또한 네트워크 사용자는 인터넷 IP 주소 부족을 해결하기 위한 네트워크 변환 기술(NAT)이나 외부 망으로부터 내부 망을 보호하기 위한 방화벽(Firewall)과 같은 Layer 3에서의 라우팅 기술을 요구하고 있다. 그러나 현재 라우터 기반의 알고리즘에서는 멀티미디어 서비스를 지원하거나 사용자의 요구조건을 충족시키면서 혼잡상황 및 불공정성을 해결하기 위한 방법을 가지고 있지 않다.

본 논문에서는 Layer 3의 라우팅 기술인 네트워크 변환 기술을 이용하여 혼잡상황을 효과적으로 해결하는 MFRED(Multiple Fairness RED) 알고리즘을 제안하였다. MFRED 알고리즘은 비반응 플로우(unresponsive flow)와 TCP와 같은 플로우(TCP-like flow) 사이에 불공정성 문제를 해결하였다. 그리고, 이 알고리즘은 혼잡 상황에 덜 민감하고 혼잡 상황의 원인이 되는 플로우로 부터 혼잡상황에 민감한 플로우를 공정하게 잘 보호하도록 동작한다.

**키워드** : 네트워크 변환기술, 큐 관리 알고리즘, 사설망, 인터넷, 공정성

**Abstract** With the recent rapid progress of Internet, the higher speed network is needed to support the exploration of ambient information from text-based to multimedia-based information. Also, demands for additional Layer 3 routing technique, such as Network Address Translator (NAT) and Firewall, are required to solve a limitation of a current Internet address space and to protect the interior network from the exterior network. However, current router-based algorithms do not provide mechanisms to solve the congestion and fairness problems, while supporting the multimedia services and satisfying the user requirements.

In this paper, to solve these problems, a new active queue management, called MFRED (Multiple Fairness RED) algorithm, is proposed. This algorithm can efficiently reduce the congestion in a router or gateway based on the Layer 3 routing technique, such as NAT. This algorithm can improve the fairness among TCP-like flows and unresponsive flows. It also works well in fairly protecting congestion-sensitive flows, i.e. fragile TCP, from congestion-insensitive or congestion-causing flows, i.e. robust TCP.

**Key words** : Network Address Translator, Active Queue Management, Private Network, Internet, Fairness

### 1. 서론

인터넷 사용이 급격하게 증가함에 따라 인터넷 트래픽은 기하급수적으로 증가하고 있으며, 인터넷 기술이 빠르게 발전함에 따라 인터넷 주소 부족을 해결하기 위한 네트워크 변환 기술이나 외부로부터 해킹 및 네트워크 보호를 제공하기 위한 방화벽과 같은 다양한 서비스

<sup>†</sup> 학생회원 : 광운대학교 전자통신공학과  
thkang@adams.kwangwoon.ac.kr  
jhkoo@adams.kwangwoon.ac.kr  
<sup>\*\*</sup> 종신회원 : 광운대학교 전자공학부 교수  
kchung@daisy.kwangwoon.ac.kr  
논문접수 : 2001년 11월 22일  
심사완료 : 2002년 5월 31일

기술들에 대한 요구가 증가되고 있다. 현재 라우터는 증가하는 인터넷 트래픽을 처리하는데 한계에 부딪치고 있으며, 추가적으로 라우터에 사용자의 요구조건을 만족시키며 인터넷 서비스를 제공하기 위한 기술들은 서비스 처리를 위한 지연시간 및 추가적인 복잡성으로 인하여 라우터에서의 혼잡상황을 가중시키고 있다[1,2].

이와 같은 네트워크 환경에서 혼잡상황을 해결하기 위한 방법으로 IETF(Internet Engineering Task Force)에서는 RED(Random Early Detection) 큐 관리 알고리즘을 권고하고 있다. 큐 관리 알고리즘은 혼잡상황이 발생하였을 경우, 패킷 손실을 인식하여 패킷을 재전송시 윈도우를 기반으로 혼잡제어를 하는 TCP 알고리즘(TCP's back-off algorithm)에 기반하여 동작하고 있다. RED 큐 관리 알고리즘은 구현이 간단하며 ECN(Explicit Congestion Notification)과 같은 특정 비트를 사용하여 혼잡상황을 전송 호스트에게 알려주기 때문에 TCP 메커니즘을 사용하는 현재의 네트워크 환경에 효과적으로 혼잡상황을 제어하고 있다[3,4,5].

하지만, 최근 급속하게 증가하고 있는 음성이나 비디오와 같은 실시간 멀티미디어 통신은 대부분 UDP 메커니즘을 사용하고 있다. 이러한 UDP 메커니즘은 네트워크의 패킷 손실에 대하여 혼잡제어 메커니즘이 없기 때문에 TCP 메커니즘을 사용하는 서비스와 UDP 메커니즘을 사용하는 서비스들간의 불공정성 문제가 발생되고 있다. 이러한 불공정성으로 인하여 큐 관리 알고리즘은 네트워크 환경에서의 혼잡 상황을 정상적으로 해결하지 못하고 있다[6][7].

이와 같이, RED 큐 알고리즘은 TCP와 같이 네트워크 상황에 적응적인 플로우(TCP-like flow)에 정상적으로 동작하며 UDP와 같이 네트워크 상황에 적응적이지 않는 플로우(unresponsive flow) 환경에서는 적용하기 어려운 문제를 가지고 있다. 또한, 네트워크에 적응적인 TCP 플로우 중에서, 일반적인 인터넷 트래픽에 비해 지연시간에 민감하며 원격의 컴퓨터를 제어하기 위한 Telnet 패킷이나 웹 트래픽에 대한 응답 패킷(ACK packet)과 같이 패킷 길이가 작으며 혼잡상황을 일으키지 않는 패킷을 구분하여 처리하지 않고 있으므로 TCP 플로우 패킷 내에서의 불공정성 문제를 가지고 있다[8]. 본 논문에서는 다양한 사용자의 요구조건을 충족시키기 위해 제공되는 부가적인 인터넷 서비스, 즉 부속한 인터넷 주소를 해결하기 위한 네트워크 주소 변환 기술이나 외부로부터 내부 망을 보호하기 위한 방화벽 기술과 같은 인터넷 기술을 이용하여 네트워크 환경에서의 혼잡상황을 효과적으로 해결하는 알고리즘을 제안하였다.

제안한 알고리즘은 인터넷 서비스의 정보를 기반으로 네트워크 상황에 적응적이지 않는 플로우와 네트워크 상황에 적응적인 플로우 사이에 불공정성 문제를 해결하고 있다. 또한 지연시간에 민감하며 원격의 컴퓨터를 제어하기 위한 패킷이나 ACK 패킷과 같이 TCP 플로우 중에서 혼잡상황을 일으키지 않는 패킷에 대해 공정성을 향상 시켰다.

본 논문은 총 5장으로 기술되어 있으며, 먼저 2장에서는 다양한 인터넷 서비스를 제공하기 위한 기술 중 하나로 많이 사용하는 네트워크 변환 기술과, 네트워크 환경에서의 혼잡상황을 효과적으로 제어하는 큐 관리 알고리즘에 대한 연구를 기술하였다, 그리고 큐 관리 알고리즘에서 불공정성을 해결하기 위해 제안된 여러 연구들에 관해 기술하였다. 3장에서는 다양한 사용자의 요구조건을 만족시키기 위해 제공되는 인터넷 기술을 이용하여 혼잡상황에서 효과적으로 동작하는 새로운 알고리즘을 제시하였다. 4장에서는 LINUX 환경의 PC 라우터에 구현한 알고리즘의 성능을 비교, 분석 하였으며, 5장에서는 결론 및 향후 연구 과제에 대해 기술하였다.

## 2. 배경 및 관련 연구

네트워크 사용자는 네트워크 속도의 한계를 넘어 보다 빠르며 다양한 모습으로 네트워크의 사용을 원하고 있으며 이러한 욕망은 다양한 모습으로 네트워크 기술에 반영되고 있다. 예를 들어 Layer 2에서 사용하던 스위치 기술이 기존 OSI 7 layer의 한계를 벗어나 각 layer의 장점을 모아 동작하는 스위치 기술에서 나타나고 있다. 즉, 라우팅을 하드웨어 방식으로 패킷을 전달(forwarding)하는 layer3 스위치, 세션 계층을 이용하여 포트에 따라 전달하는 layer 4의 웹 스위치(WebSwitch), 어플리케이션의 중요도에 따라 동작하는 layer 7의 앱스 스위치(AppSwitch)와 같은 기술이 사용되고 있다.

본 논문을 쓰게 되는 배경은 사용자의 부가적인 서비스를 위해 사용되는 Layer 3의 네트워크 변환 기술의 장점을 이용하여 보다 효율적으로 큐 관리 알고리즘에 적용시키고자 하였다.

### 2.1 네트워크 주소 변환 기술(NAT)

사설 망(Private Network)은 개인이나 기업이 사적인 목적으로 구축한 네트워크로 내부적인 규정에 따라 관리되거나 운영되는 네트워크이다. 사설망은 네트워크 주소 변환 기술을 사용하여 외부의 공인 망과 연결하고 있으며 사설 망에서는 주고 받는 정보들을 해당 조직의 사용자가 접근할 수 있도록 보호하고 있다. 또한 IP 주소 사용이나 네트워크 관련 규칙들을 그 조직 내부의

임의대로 정의할 수 있는 망에서 사용되고 있다. 네트워크 주소 변환 기술은 사실 망을 기반으로 기존의 네트워크의 IP 주소 체계를 그대로 유지하면서 부족한 IP 주소 문제를 해결하기 위한 방법을 제공한다. 이 기술은 소수의 공인된 인터넷 주소를 이용하여 다수의 사용자에게 인터넷 서비스를 제공하기 위한 인터넷 서비스 기술 중의 하나이다.

네트워크 변환 기술은 그림 1과 같이 Layer 2와 Layer 3 사이에 위치하여 동작한다. 동작 방법은 입력되는 모든 패킷에 대해서 인터넷 주소 변환 과정을 거친 후 상위 계층인 Layer 3에 전달하며, Layer 3에서는 일반적인 라우터와 동일하게 인터넷 주소의 가장 긴 프리픽스 매칭(longest prefix matching)을 이용하여 패킷을 라우팅한다.

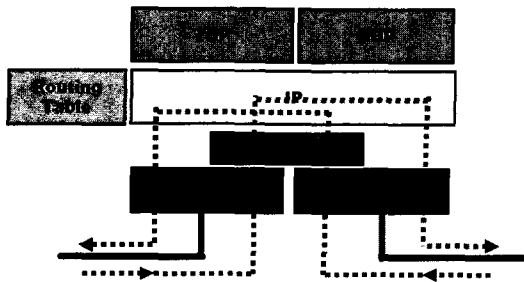


그림 1 NAT의 프로토콜 계층 및 동작 방법

네트워크 주소 변환 기술은 사실 망에서 생성한 플로우와 이에 대한 응답으로 공인 망에서 생성하는 응답 플로우(ACK flow)를 연결시키기 위해 표 1과 같은 NAT 테이블을 사용한다.

표 1 NAT 테이블

10.10.1.1 : 1025	128.134.x.120 : 1067	211.32.119.151 : 1049	900
10.10.1.2 : 2100	128.134.x.120 : 1068	128.134.70.1 : 80	671
10.10.1.10 : 1600	128.134.x.120 : 1069	210.113.67.131 : 4050	455

네트워크 주소 변환 기술은 사실 망에 생성한 패킷을 NAT 라우터에서 사용하는 소수의 공인 인터넷 주소로 변환하여 특정 포트번호로 외부의 공인 인터넷에 전달하며, 변환에 사용되었던 플로우 정보를 NAT테이블에 저장한다. 외부 공인 망에서 서비스되는 플로우는 NAT 라우터에서 NAT 테이블의 특정 포트 번호, 수신자의 주소(source address), 이용된 프로토콜을 바탕으로 사실 망 호스트에게 전달하여 준다.

이와 같이 네트워크 변환 기술은 기존의 인터넷 체계를 그대로 유지하면서 인터넷 주소 부족을 해결하고 있다. 또한 NAT의 플로우 연결은 NAT 테이블을 기반으로 연결하므로 네트워크 보호를 위한 방화벽 기능을 제공하고 있다.

그러나 사용자에게 다양한 서비스를 제공하기 위한 네트워크 주소 변환은 오히려 라우터에 복잡도를 증가시키므로 네트워크 환경에 혼잡상황을 가중시키고 있다.

2.2 RED(Random Early Detection) 알고리즘

1992년 Floyd가 제안한 RED 알고리즘은 라우터와 같은 네트워크 코어(core)에서 네트워크 상황에 맞게 큐를 관리함으로써 네트워크 환경에서의 혼잡 상황을 해결하는 알고리즘이다. RED 큐 알고리즘은 라우터에서 처리되는 큐 크기를 일정한 평균값을 유지하도록 관리하여 짧은 주기 동안 트래픽이 집중되어 발생하는 일시적인 혼잡 상황과 장시간 트래픽이 집중되어 발생하는 지속적인 혼잡 상황을 효과적으로 해결하고 있다.

RED 동작은 EWMA(Exponential Weighted Moving Average) 기반으로 그림 2와 같이 라우터의 평균 큐 크기를 계산하여 큐의 최대, 최소한계치에 따라 패킷을 폐기하는 동작을 한다.

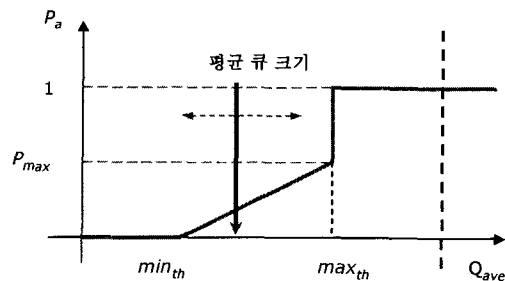


그림 2 RED 알고리즘의 패킷 폐기 확률 분포

평균적인 큐 크기가 최소 한계치보다 작을 경우 모든 패킷을 서비스하며, 평균 큐 크기가 최대 한계치보다 큰 경우 모든 패킷을 폐기한다. 평균 큐 크기가 최소 한계치와 최대 한계치 사이에 있을 경우에는 임의적으로 특정 확률 값에 따라 패킷을 폐기함으로써 중단 호스트에게 혼잡의 상황을 알려준다.

RED는 평균 큐 크기를 기반으로 혼잡상황에 대하여 처리하기 때문에 그림 3과 같이 혼잡상황이 발생하는 경우, 기존의 Drop-tail 방식에서 발생되던 full queue와 lock out과 같은 현상이 없으므로 향상된 전송률을 보여주고 있다.

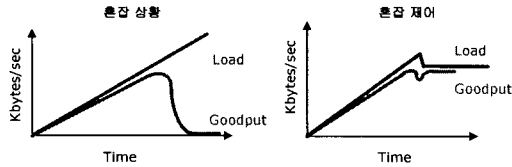


그림 3 혼잡 상황과 혼잡 제어 상황의 비교

RED는 패킷 손실을 인식하여 재전송시 윈도우 기반으로 혼잡제어를 하는 TCP의 알고리즘(TCP's back-off algorithm)에 기반하여 동작하고 있으므로 패킷 손실에 반응을 하지 않는 UDP 플로우와 같은 메커니즘에 대하여 적용하지 못한다는 문제를 가지고 있다. 즉 UDP 플로는 혼잡상황으로 인하여 패킷이 손실되었을 경우, 혼잡상황에 대해 전송 호스트는 혼잡제어 동작을 하지 않으므로 네트워크에 적응적인 TCP 플로우와 경쟁으로 인하여 불공정성을 발생시키고 있다.

간단히 두 메커니즘 사이에서 발생하는 불공정성을 알아보기 위해서 TCP 플로우와 UDP 플로우를 동시에 전송하였을 경우, 그림 4와 같이 네트워크에 적응적이지 않는 UDP 플로우가 네트워크의 대역폭을 대부분 점유함으로써 TCP 플로우가 정상적으로 서비스되지 못하는 불공정성을 보여주고 있다.

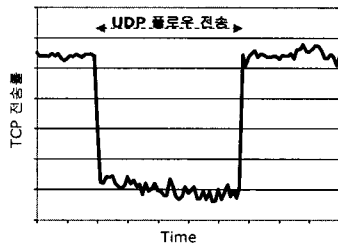


그림 4 UDP 영향에 의한 TCP의 전송률 변화

또 다른 RED의 불공정성 문제로, RED는 TCP 플로우 중에서도 패킷 크기가 크며 대역폭을 많이 사용하는 패킷과 패킷 크기가 작으며 지연 시간에 민감한 패킷을 구분하지 않고 동일하게 폐기하기 때문에 불공정성 및 비효율적 문제를 가지고 있다.

### 2.3 RED의 불공정성을 해결하기 위한 관련 연구

혼잡상황에 적응적이지 않는 플로우(unresponsive flow)에 의해 발생하는 불공정성 문제를 해결하기 위해, 플로우 단위로 공정성을 판단하는 FRED(Flow based RED), UDP 플로우의 트래픽 양을 특정 한계 값으로 제한하여 공정성을 향상한 CBT(Class Based Threshold),

그리고 CBT의 특정 한계 값을 네트워크 환경에 맞게 수정한 DCBT(Dynamic CBT)가 제안되었다[9,10,11]. 또한 TCP 플로우 내에서 발생하는 불공정성 문제를 해결하기 위해, TCP 플로우의 RTT값을 기반으로 공정성을 판단하는 LQD(longest queue drop)가 제안되었다[12].

FRED 알고리즘은 플로우 단위(per-flow)로 패킷을 관리하여 특정 플로우의 큐 크기가 공정성의 한계치를 넘게 되면 특정 플로우에 대해 보다 엄격하게 큐를 관리함으로써 RED에서 발생하는 불공정성을 해결하였다. 그러나 layer 2인 큐에서 모든 플로우에 대해 플로우 단위로 패킷의 공정성을 계산하기 때문에 구현의 어려움 및 알고리즘의 높은 복잡도를 가지는 문제점이 있다.

CBT 알고리즘은 FRED에서 플로우마다 관리되는 복잡도를 줄이기 위하여 모든 플로우를 세가지 등급의 플로우로 나누었다. 즉, UDP 등급의 플로우를 큐의 정해진 한계값(threshold)내에서 사용하도록 큐의 크기를 제어함으로써 TCP와 UDP 사이에서의 공정성 및 FRED 알고리즘의 복잡도 문제를 해결하고자 하였다.

또한 DCBT 알고리즘은 CBT 알고리즘의 정해진 한계값을 평균 큐 크기에 따라 설정함으로써 CBT 알고리즘의 공정성 한계를 개선하였다.

그러나 CBT, DCBT는 에지 라우터에서 플로우마다 등급을 나누어 표시하거나 전송 호스트에서 미리 플로우에 대한 등급을 나누어 전송을 해야 하는 문제를 가지고 있었다.

이와는 다르게 TCP 플로우 내에서 발생하는 공정성을 개선하기 위한 방법으로 LQD가 제안되었다. LQD는 RTT 값으로 플로우를 구분하여 대역폭을 많이 사용하는 플로우는 공정성을 판단에 의해 큐의 버퍼 크기를 적게 적용함으로써 대역폭을 적게 사용하는 TCP 플로우 사이의 공정성을 향상시켰다.

LQD는 RTT 값과 큐를 많이 점유하는 플로우에 대해 탐색 및 정렬(searching or sorting)을 해야 하는 복잡성 문제가 있다.

따라서 본 논문에서는 네트워크의 혼잡상황이 발생하였을 경우, 적은 복잡도를 가지면서도 불공정성 문제를 해결하기 위한 새로운 큐 관리 알고리즘을 제안하고자 한다.

### 3. MFRED 알고리즘

본 논문에서는 앞에서 기술한 다양한 네트워크 환경에서 발생될 수 있는 불공정성 문제를 해결하기 위해 기존의 큐 관리 알고리즘의 문제를 해결한 새로운 큐 관리 알고리즘인 MFRED(Multiple Fairness Random

Early Detection)를 제안한다. MFRED는 다양한 인터넷 서비스를 사용하며 혼잡 상황이 발생하는 게이트웨이나 라우터에서 다음과 같은 기능을 지원하기 위해 설계되었다.

MFRED 알고리즘을 사용하는 목적은 다음과 같다.

- 트래픽이 집중되는 라우터나 게이트웨이에서 네트워크 주소 변환 기술과 같은 인터넷 기술로 인하여 가중되는 혼잡 상황을 효과적으로 제어한다.
- 네트워크 상황에 적응적이지 않는 플로우와 네트워크 상황에 적응적인 플로우 사이에 불공정성 문제를 해결한다.
- TCP 플로우 중에서 혼잡 상황을 일으키지 않으며 지연 시간에 민감한 패킷 즉, ACK 패킷이나 원격 제어(remote control) 패킷을 보다 공정하게 처리한다.

**3.1 MFRED의 동작 방법**

본 논문에서 제안하는 MFRED는 NAT 테이블에서 사용하는 Layer 3의 플로우 정보를 기반으로 TCP 플로우 수 및 UDP 플로우 수를 계산한다. 혼잡 상황이 발생할 경우, Layer 3의 플로우 수를 Layer 2의 큐 알고리즘에 적용함으로써 네트워크에 적응적인 플로우와 네트워크에 적응적이지 않는 플로우 사이의 불공정성을 해결한다. 또한 TCP 플로우 중에서 혼잡 상황을 일으키지 않으며 지연 시간에 민감한 패킷에 대해 TCP 패킷의 평균 크기를 기반으로 패킷 폐기 확률을 낮춤으로써 TCP 플로우 내에서 공정성을 보장한다.

MFRED는 그림 5와 같이 NAT 테이블에서 사용하는 정보를 기반으로 TCP, UDP의 플로우 수를 얻는다. Layer 3의 플로우 정보는 Layer 2에 위치한 큐 관리 알고리즘에 제공하여 플로우 수를 기반으로 패킷에 대한 폐기 확률을 결정한다.

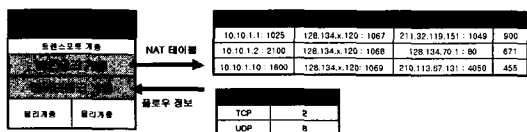


그림 5 NAT 테이블에서 플로우 정보를 얻는 과정

그림 6은 MFRED의 동작 알고리즘을 보여주고 있다.

MFRED 큐 관리에서 사용하는 TCP, UDP의 플로우 수는 Layer 2에 저장되는 패킷 수가 정해진 임계 값 이상 되었을 경우에 NAT 테이블을 통하여 플로우 정보를 읽어 들인다. Layer3의 플로우 정보를 Layer 2에 참조하는 방법은 일정 시간마다 정보를 읽어 들이는 방법과 임의의 패킷수 마다 정보를 읽어 들이는 방법이 있

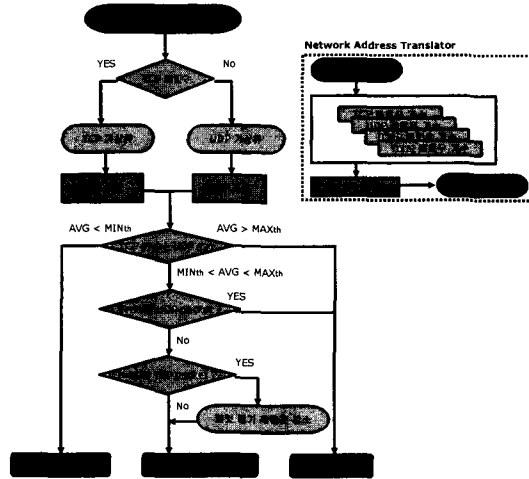


그림 6 MFRED의 동작 알고리즘

다. 본 논문에서는 큐에 입력되는 패킷수가 많아짐에 따라 greedy UDP를 보다 정확하게 판별하며, fragile TCP를 판별시 패킷 길이를 계산하기 위해 후자의 방법을 사용하였다. 실험 환경상 greedy 플로우를 고려하여 50개의 패킷을 임계 값으로 설정하였다.

MFRED는 기존 RED와는 다르게 하나의 물리적인 큐 위에 두 개의 가상 큐를 가지는 구조로 되어서 각각 TCP 가상 큐와 UDP 가상 큐를 가지고 있으며, 혼잡 상황이 발생하였을 경우 각각 평균적인 TCP와 UDP 큐 크기를 알아낼 수 있다.

MFRED의 동작은 일반적으로 혼잡상황이 발생하였을 경우, TCP가 사용하는 큐 크기와 UDP가 사용하는 큐 크기의 합에 대한 평균 큐 크기가 최소 한계치보다 작게 되면 그대로 패킷을 그대로 전달하며, 최대 한계치보다 크게 되면 실제 혼잡 상황이 발생하는 것을 막기 위해 모든 패킷을 폐기한다.

TCP 평균 큐의 크기와 UDP 평균 큐의 크기를 더한 큐의 평균값(TCP Qavg + UDP Qavg)이 최소 한계치(MINth)와 최대 한계치(MAXth) 사이에 있을 경우, NAT에서 제공한 플로우 연결 수를 기반으로 TCP의 공정성에 어긋나는 UDP의 패킷을 폐기함으로써 혼잡제어를 하지 않는 UDP 플로우를 제어하였다.

Greedy UDP를 구별하는 식은 참고 문헌 [11]의 Tagged UDP를 판단하는 식을 기반으로 다음과 같이 표현할 수 있다. 식 (1)에서는 평균 큐의 크기가 최소 한계 값(MINth) 이상이 되면 UDP 패킷을 혼잡상황을 유발시키는 패킷이라 가정하여 Layer3의 NAT 테이블에서 읽어 들인 플로우 정보에 따라 greedy UDP를 판

별하여 패킷을 폐기한다. 그러므로 식 (1)과 같이 혼잡 상황시 큐에 저장되어 있는 UDP의 평균 큐 크기를 비교하여 UDP 패킷을 폐기한다.

Greedy udp :

$$udp_{queue\ size} > \frac{udp\ flow\ number}{(udp\ flow\ number + tcp\ flow\ number)} \times (tcp_{queue\ size} + udp_{queue\ size}) \quad (1)$$

Fragile TCP를 구별하는 식은 다음 식 (2)와 같이 전체 TCP가 사용하는 큐를 패킷 수로 나누어 혼잡 상황시 큐에 저장되어 있는 TCP의 평균 길이보다 작은 길이의 TCP 패킷에 대한 폐기 확률을 감소시켰다. TCP 패킷 길이는 IP 헤더에 있는 패킷 길이(HLEN)에 따라 적용시켰으며 본 논문의 실험에서 패킷 폐기 확률을 80%로 감소시켰다.

Fragile tcp :

$$tcp\ packet\ size < \frac{tcp_{queue\ size}}{tcp\ packet\ number} \quad (2)$$

#### 4. MFRED 시험 망 구성 및 성능평가

##### 4.1 시험망 구성 및 시험 방법

본 논문에서의 MFRED 알고리즘의 성능을 알아보기 위해 제안한 알고리즘은 LINUX[13]에서 구현하였으며 실험 환경을 그림 7과 같이 구성하였다. S1에서 전송하는 3개의 TCP 플로우를 NAT 라우터를 거쳐 각각 R1, R2, R3에 수신하게 하였으며, S2에서 전송하는 1개의 UDP 플로우는 NAT 라우터를 거쳐 R4에 수신하도록 하였다. 측정 도구로는 트래픽 생성 및 대역폭을 측정하기 위해 iperf[14]를 사용하였으며, 지연시간을 측정하기 위해 netperf[15]를 사용하였다. 또한 플로우 간의 전송률을 그래프적으로 모니터링 하기 위해 ttt[16]를 사용하였다.

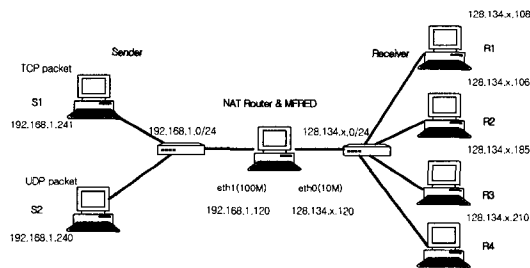


그림 7 실험 환경

네트워크 변환 기술에서의 플로우 연결은 NAT 테이블 기반을 동작한다. 즉, 네트워크 변환 기술은 외부 공

인 망에서 사설 망의 호스트에게 패킷을 전달시키지 않는 방화벽 기능을 제공하므로 본 논문에서는 사설 망에서 외부의 공인 망 방향으로 혼잡 상황이 발생하도록 하였다.

##### 4.2 MFRED의 성능 평가

제안한 MFRED의 전송률에 대한 비교를 하기 위해 주기가 20초, 버퍼 사이즈가 8k 바이트, 송수신 패킷 크기가 64바이트인 3개의 TCP 플로우를 연결하여 netperftools를 이용하여 측정하였다. 또한 제안한 MFRED의 지연 시간에 대한 비교를 하기 위해 주기가 20초이고 송수신 패킷 크기가 64바이트인 1개의 UDP 플로우를 연결하여 측정하였다.

표 2와 표 3은 NAT를 사용하지 않았을 경우와 NAT를 사용하였을 경우 비교하여 지연 시간의 증가를 보여주고 있다. NAT에 RED를 사용하였을 경우는 RED를 사용하지 않았을 경우와 비교하여 지연 시간 감소 및 전송률이 증가함을 보여주고 있다. 그러나 MFRED의 경우에는 NAT에 RED를 적용한 경우보다 복잡성에 따른 지연 시간 및 전송률의 차이가 거의 없음을 보여주고 있다.

표 2 TCP 플로우 전송률 비교

	262142	131070	64	20.00	3.70
FIFO	262142	131070	64	20.00	3.71
NAT & FIFO	262142	131070	64	20.00	3.75
NAT & RED	262142	131070	64	20.00	3.70
MFRED	262142	131070	64	20.00	3.70

표 3 TCP 플로우 지연 시간 비교

	131070	131070	64	64	20.00	1750.69
FIFO	131070	131070	64	64	20.00	1750.77
NAT & FIFO	131070	131070	64	64	20.00	1653.27
NAT & RED	131070	131070	64	64	20.00	1671.66
MFRED	131070	131070	64	64	20.00	1671.66

그러므로 사용자의 요구조건을 충족시키기 위해 사용하는 네트워크 변환기술은 기존의 라우터에 패킷을 처리하기 위한 지연시간 발생 및 부가적으로 혼잡상황을 유발하고 있음을 확인 할 수 있다. 또한 본 논문에서 제안한 MFRED를 사용하였을 경우, 사용자의 요구조건을 충족시키면서 혼잡상황을 효과적으로 제어하고 있음을 확인할 수 있다.

##### 4.3 Unresponsive flow와 TCP-like flow간의 공정성 평가

혼잡상황이 발생하는 경우 공정성을 판단하기 위해 윈도우 크기가 16KBbyte인 2개의 TCP 플로우와

32KByte인 FTP 플로우를 지속적으로 백그라운드 트래픽으로 설정한 후, UDP 플로우를 8M로 전송하였다. MFRED의 공정성을 알아보기 위해 그림 8과 같이 NAT 라우터의 인터페이스(eth0)에서 ttt 툴을 이용하여 플로우에 따른 전송률을 측정하였다.

RED를 사용하였을 경우 그림 8과 같이 혼잡상황에 반응을 하지 않는 UDP 플로우의 전송 시점부터 라우터의 대역폭을 거의 차지하고 있음을 보여주고 있다. 또한 TCP 플로우 내에서 지속적으로 대용량 데이터를 전송하는 FTP 플로우와 일시적으로 전송되는 웹 트래픽과 같은 TCP 플로우와의 불공정성을 보여주고 있다.

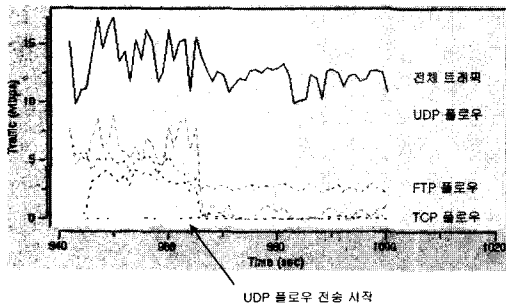


그림 8 RED를 사용한 경우 플로우에 따른 전송률

그러나 MFRED를 사용하였을 경우, 그림 9와 같이 NAT 테이블에서 제공하는 플로우의 연결수에 따라 혼잡상황에 반응을 하지 않는 UDP 플로우를 제어함으로써 네트워크에 적응적인 TCP 플로우와 네트워크에 적응적이지 않는 UDP 플로우에 따른 불공정성 문제를 해결하고 있다. 또한 TCP 플로우 내에서 대용량의 데이터를 지속적으로 전송하는 FTP 플로우와 크기가 작으며 일시적으로 전송되는 패킷에서 발생하는 TCP 플로우 사이에서의 불공정성 문제를 해결하고 있음을 보여주고 있다.

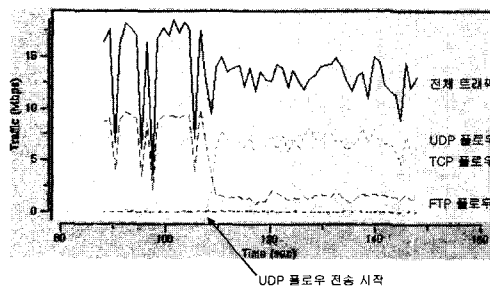


그림 9 MFRED를 사용한 경우 플로우에 따른 전송률

#### 4.4 TCP 플로우 내에서의 공정성 평가

혼잡 상황이 발생하였을 경우 TCP 플로우 내에서의 패킷에 대한 공정성을 판단하기 위해, Telnet에서 원격의 컴퓨터를 제어하기 위한 패킷이나 ACK 패킷과 같이 패킷 길이가 작으며 혼잡상황을 일으키지 않는 패킷을 전송하여 그에 대한 전송률 및 지연시간을 알기 위해, 이와 유사한 트래픽 특성의 패킷을 전송하여 전송률 및 지연시간을 측정하였다.

표 4와 표 5는 TCP 플로우 중에서 패킷 길이가 ACK 패킷과 유사하게 메시지 길이가 1byte의 플로우를 연결하여 그에 따른 전송률 및 지연시간을 대해 비교하고 있다. 혼잡상황 시 패킷을 구분하지 않고 처리하는 RED의 경우, 지연 시간에 민감하며 대역폭을 거의 사용하지 않는 패킷(64bytes)을 대역폭을 많이 사용하는 패킷(1500bytes)과 동일하게 처리하므로 패킷을 구별하여 처리하는 MFRED에 비해 지연시간 증가 및 낮은 전송률을 보여주고 있다.

표 4 메시지 길이가 1bytes인 ACK 패킷 전송 시 전송률 비교

Queueing	Packet Size	Rate	Delay	Throughput	Efficiency
NAT & FIFO	9216	41600	1	26.58	0.57
NAT & RED	9216	41600	1	26.50	0.59
MFRED	9216	41600	1	25.53	0.68

표 5 메시지 길이가 1bytes인 ACK 패킷 전송 시 지연시간 비교

Queueing	Packet Size	Rate	Delay	Throughput	Efficiency	
NAT & FIFO	131070	131070	1	1	20.00	12.32
NAT & RED	131070	131070	1	1	20.00	12.38
MFRED	131070	131070	1	1	20.00	11.91

그러므로 혼잡 상황시 MFRED를 사용함으로써, 원격의 시스템을 제어하는 Telnet 패킷이나 웹 트래픽의 응답 패킷인 ACK 패킷에 대해 보다 낮은 지연 시간 감소 및 효율적인 성능을 제공하고 있음을 확인할 수 있다.

#### 5. 결론 및 향후 과제

인터넷 기술이 급속하게 발전함에 따라 네트워크 사용자는 텍스트 환경에서 벗어나 음성과 영상 전송을 위한 멀티미디어 환경의 서비스를 보다 빠르게 사용하기를 원하고 있으며 네트워크 사용자는 기존의 네트워크의 한계를 넘어 보다 다양한 방식의 네트워크 사용을 요구하고 있다. 이와 같은 요구사항을 만족시키기 위해 OSI 7 Layer의 제한을 벗어나 하드웨어로 동작하는 layer 3 스

위치, Layer 4의 웹 스위치(WebSwitch), layer 7의 앱스 스위치(AppSwitch)와 같은 네트워크 기술이 사용되고 있다. 또한 이와는 별개로 네트워크 사용자는 기존에 제공되는 인터넷 서비스의 한계를 넘어 보다 다양한 방식의 인터넷 사용을 요구하고 있다. 이와 같은 부가적인 인터넷 서비스를 제공하기 위한 기술로써 네트워크 주소 변환 기술이나 방화벽과 같은 인터넷 서비스가 사용되고 있다.

그러나 라우터는 다른 기종의 네트워크를 연결하여 네트워크의 패킷을 목적지로 전달시켜주는 장치로써 각 호스트에서 발생하는 트래픽이 라우터에 집중으로 인하여 네트워크 코어에서 혼잡상황으로 발생하는 병목현상의 주 원인이 되고 있다. 이와 같은 라우터의 혼잡상황을 효과적으로 제어하기 위한 방법으로 큐 관리 알고리즘이 있으나 큐 관리 알고리즘은 네트워크 사용자의 다양한 요구를 충족시키면서 추가적으로 발생하는 혼잡상황 및 불공정성을 해결하기 위한 명시적인 방법을 가지고 있지 않다.

그러므로 본 논문에서는 사용자의 다양한 요구조건을 충족시키기 위해 사용되는 인터넷 기술을 이용하여 네트워크의 혼잡상황을 효과적으로 해결하는 MFRED(Multiple Fairness Random Early Detection) 알고리즘을 제안하였다. MFRED 알고리즘은 Layer 2의 큐 관리 알고리즘에서 Layer 3에서 동작하는 네트워크 주소 변환 기술을 이용한다. MFRED는 네트워크 변환 기술의 플로우 정보를 기반으로 네트워크 상황에 적응적이지 않는 플로우와 네트워크 상황에 적응적인 플로우 사이에서 발생하는 불공정성 문제를 해결하였다. 또한 패킷의 크기가 작으며 지연 시간에 민감한 원격 제어(remote-control) 패킷이나 응답(ACK) 패킷과 같이 TCP 플로우 내에서 혼잡상황을 일으키지 않는 패킷에 대해 보다 공정하게 처리함으로써 보다 나은 성능을 제공하였다.

결론으로써 저자는 MFRED 알고리즘을 각 Layer의 장점을 이용한 다양한 스위치 기술과 같이 Layer 3의 네트워크 변환 기술의 특징을 이용하여 네트워크의 혼잡상황을 효과적으로 제어하는 큐 관리 알고리즘으로 의의를 두고자 한다.

마지막으로, MFRED는 TCP의 가상 큐의 크기를 기반으로 UDP의 공정성을 판단하고 있다. 그러므로 TCP 가상 큐의 크기로써 TCP 플로우에 대한 정보를 제공하지 못하는 경우와 함께 TCP 플로우 중에서 RTT값에 의해서 발생하는 불공정성에 관한 연구가 이루어져야 할 것이다. 또한 인터넷 사용자의 다양한 서비스 요구조건을 만족시키기 위해서 보다 나은 큐 관리 알고리즘에 관한 연구도 필요하겠다.

## 참 고 문 헌

- [1] Senie, D., "NAT Friendly Application Design Guidelines," February 1999. [<http://www.ietf.org/internet-drafts/draft-ietf-nat-guide-01.txt>].
- [2] Newman, D., "Benchmarking Terminology for Firewall Performance," IETF RFC 2647. August 1999.
- [3] Stevens, W., "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," IETF RFC2001, January 1997.
- [4] Floyd, S., "Congestion Control Principles," IETF RFC 2914, September 2000.
- [5] Ramakrishnan, K.K., Floyd, S., Black, D., "The Addition of Explicit Congestion Notification (ECN) to IP," IETF RFC3168, September 2001.
- [6] Braden et al, B., "Recommendations on Queue Management and Congestion Avoidance in the Internet," IETF RFC2309, April, 1998.
- [7] Hong, D., Albuquerque, C., Oliveira, C., Suda, T., "Evaluating the Impact of Emerging Streaming Media Applications on TCP/IP Performance," IEEE Communication Magazine, April 2001.
- [8] Floyd, S., Fall, K., "Promoting the Use of End-to-End Congestion Control in the Internet," IEEE/ACM Transaction on Networking, August 1999.
- [9] Lin, D., Morris, R., "Dynamic of Random Early Detection," SIGCOMM, 1997.
- [10] Parris, M., Jeffay, K., Smith, F., "Lightweight Active Queue Management for Multimedia Networking," Multimedia Computing and Networking, SPIE Proceeding Series, vol 3020, San Jos CA, January 1999.
- [11] Jae, C., Claypool, M., "Dynamic-CBT, Better Performing Active Queue Management for Multimedia Networking," In Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), Chapel Hill, NC, USA, June 2000.
- [12] Suter, B.T.V., Lakshman, Siliadis, D., and Choudhury, A.K., "Design Considerations for Supporting TCP with Per-flow Queuing," Proceeding of IEEE INFOCOM 1998.
- [13] Red Hat LINUX (kernel version 2.4-2) [<http://www.redhat.com>].
- [14] IPERF (Internet bandwidth Performance) [<http://dst.nlanr.net/Projects/Iperf/>].
- [15] NETPERF(Network Performance Benchmark) [<http://www.netperf.org/netperf/NetperfPage.html>].
- [16] TTT(Tele Traffic Tapper) [<http://www.csl.sony.co.jp/person/kjc/programs.html>].
- [17] 구자현, 송병훈, 정광수, 오승준, "라우터에서의 동적



인 혼잡 제어를 위한 새로운 큐 관리 알고리즘”, 한국정보과학회, 제 28권 제 4호, 2001.12.

- [18] Peda, P., Seddigh, N., Nady, B., "The Dynamics of TCP and UDP interaction in IP-QoS Differentiated Services Network," The 3<sup>rd</sup> Canadian Conference on Broadband Research, November 1999.
- [19] Feng, W., Kandlur, D., Saha, D., Shin, K., "Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fairness," in Proc. of INFOCOM 2001, April 2001.



강 태 형

2000년 광운대학교 전자통신공학과 학사.  
2002년 ~ 현재 광운대학교 전자통신공학과 석사 과정. 관심분야는 컴퓨터 통신, 인터넷 QoS, 멀티미디어통신

구 자 현

정보과학회논문지 : 정보통신  
제 29 권 제 4 호 참조

정 광 수

정보과학회논문지 : 정보통신  
제 29 권 제 3 호 참조