

Core-Stateless망에서의 공정한 대역폭 할당 방식

김 문 경* · 박 승 섭**

요 약

공정한 전송률을 보장하고 지연을 줄이기 위해 인터넷상에서 많은 흐름단위 스케줄링 알고리즘이 제안되어졌으며, 효율성이 필요한 공정한 대역폭 할당을 위한 공정큐잉 알고리즘이 설계되었다. 그러나 흐름단위 처리를 기본으로 하는 알고리즘은 전송률 상태, 버퍼 관리, 패킷 스케줄링 등의 기능을 필요로 하며, 이와 같은 복잡성으로 인해 구현하는데 비용이 많이 드는 문제점이 있다.

따라서 본 연구는 효율적인 비용으로 라우터를 구현하기 위해, FNE(Flow Number Estimation)를 기반으로 한 CS-FNE(Core Stateless FNE) 알고리즘을 제안하며, 다양한 네트워크 환경과 트래픽 소스를 사용해서 CSFQ(Core Stateless Fair Queueing), FRED(Fair Random Early Detection), RED(Random Early Detection), DRR(Dynamic Round Robin) 등과 같은 알고리즘을 CS-FNE와 함께 평가하였다. 모의 실험 결과로, 본 연구에서 제안된 방식이 다른 알고리즘보다 적절히 공정한 대역폭 할당을 나타내었으며, 흐름단위 기반 큐잉 메커니즘보다 더 단순하고, 쉽게 구현될 수 있음을 보였다.

키워드 : 공정성, 스케줄링, CSFQ, FRED, RED, DRR, CS-FNE

Fair Bandwidth Allocation in Core-Stateless Networks

Mun-Kyung Kim* · Seung-Seob Park**

ABSTRACT

To provide the fair rate and achieve the fair bandwidth allocation, many per-flow scheduling algorithms have been proposed such as fair queueing algorithm for congestion control. But these algorithms need to maintain the state, manage buffer and schedule packets on a per-flow basis; the complexity of these functions may prevent them from being cost-effectively implemented.

In this paper, therefore, to acquire cost-effectively for implementation, we propose a CS-FNE(Core Stateless FNE) algorithm that is based on FNE(Flow Number Estimation), and evaluated CS-FNE scheme together with CSFQ(Core Stateless Fair Queueing), FRED(Fair Random Early Detection), RED(Random Early Detection), and DRR(Dynamic Round Robin) in several different configurations and traffic sources. Through the simulation results, we showed that CS-FNE algorithm can allocate fair bandwidth approximately than other algorithms, and CS-FNE is simpler than many per-flow basis queueing mechanisms and it can be easily implemented.

Key Words : Fairness, Scheduling, CSFQ, FRED, RED, DRR, CS-FNE

1. 서 론

최근 통신의 발달로 인터넷 트래픽 급속히 증가해가고 있다. 단순히 비실시간 데이터 전송 위주였던 인터넷 서비스에 멀티미디어 음성 및 실시간 화상 데이터와 온라인 게임, P2P, 원격교육, 전자 상거래 등 다양한 서비스가 생겨남에 따라 네트워크상의 공정성, 확장성, 보안성 등 여러 방면에서의 보다 향상된 네트워크를 요구하게 된다.

인터넷망의 고속화에 따른 실시간 서비스 품질을 위해 IETF(Internet Engineering Task Force)에서 MPLS

(Multi-Protocol Label Switching) 기술을 정의하고 있다. 그러나 모든 망을 MPLS 라우터로 변환 하는 것은 네트워크 방식이 변해야 하므로 비용이 많이 든다. 또한 중계역할을 하는 노드가 같은 방식으로 대역폭을 관리하면서 TCP/IP 데이터를 처리하는 것도 많은 비용이 든다. 그래서 중심 노드(Core Node)와 경계 노드(Edge Node)의 역할을 따로 두어 처리하는 Core-Stateless망에 대한 연구가 이루어지고 있다. 대표적으로 Core-Stateless망에서의 대역폭 공정성 연구에 대한 CSFQ(Core-Stateless Fair Queueing)알고리즘이 있다[1].

인터넷 트래픽은 매년 증가하고 있다. 네트워크는 자원을 효율적으로 제어할 필요가 있으며 이러한 인터넷 혼잡제어는 종단간 전송에 있어서 안전성과 효율성을 요구하며 대역폭 역시 공정하게 할당될 필요가 있다. 공정한 대역폭 할당을 위해 일반적으로 흐름단위(per-flow) 큐잉 제어 메커니

* 본 논문은 BK21기금의 지원에 의해 연구되었음.

† 준 회원 : 부경대학교 대학원

** 정 회원 : 부경대학교 전자컴퓨터정보통신공학부 교수(교신저자)

논문접수 : 2005년 2월 1일, 심사완료 : 2005년 7월 11일

즘이 많이 연구되어 왔다. 대표적으로 FQ(Fair Queueing), WFQ(Weighted Fair Queueing), 흐름단위 폐기에 중점을 둔 FRED(Flow Random Early Drop) 등 있다[2], [3], [4].

하지만 이러한 흐름단위 메커니즘은 흐름단위 상태 관리 유지와 패킷 분류를 요구하는 큐잉 기법이기에 때문에 많은 흐름을 가진 고속망에서 적합하지 않으며 구현하는데 복잡하다는 문제점이 있다.

최근 이러한 복잡성의 문제점을 해결하기 위해 여러 가지 프레임워크와 메커니즘의 제안되어 왔는데, 특히 Stoica는 Core-Stateless망에서 CSFQ(Core Stateless Fair Queueing)를 제안하였으며, Cao는 RFQ(Rainbow Fair Queueing)를 제안하였다[1][5]. 두 방식의 차이점은 크게 CSFQ 는 패킷을 흐름단위로 처리하고 흐름에 따라 label을 할당한다. RFQ는 패킷 흐름 당 두지 않고 패킷 흐름의 평균속도에 따라 color label을 할당한다. 그러나 들어오는 패킷의 라벨을 지수분포로 계산하기 때문에 복잡성을 줄일 필요가 있다.

최근 제안된 Li의 FNE(Flow Number Estimation) 큐잉기법은 해쉬 함수를 이용해서 큐의 패킷수를 계산하는 방법을 단순하게 구현한 방법이다[6].

본 연구는 이 알고리즘을 이용해서 Core-Stateless망에서 큐잉의 복잡성을 줄이고 공정하게 대역폭을 할당하는 방식을 구현하기 위해, FNE(Flow Number Estimation) 기법을 Core-Stateless망에 적용하고 망에 맞게 구현하였다. 이를 본 연구는 CS-FNE (Core Stateless FNE)로 명명하고 여러 알고리즘과 비교 분석하였다.

본 논문의 구성은 서론에 이어, 2장 CS-FNE 알고리즘 제안 및 설명에서는 Core-Stateless망과 FNE(Flow Number Estimation) 알고리즘, CS-FNE에 대해서 설명하고, 3장에서는 모의실험의 망과 파라미터 설정에 대해서 설명하며, 결과 분석을 한다. 마지막 4장은 본 논문의 결론이다.

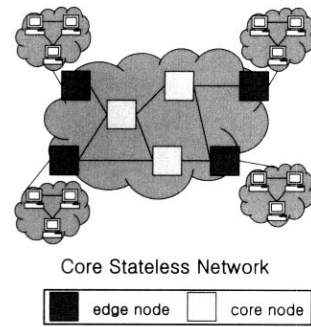
2. CS-FNE 알고리즘 제안 및 설명

본 장은 Core-Stateless망을 설명하고, FNE(Flow Number Estimation) 기법을 Core-Stateless망에 적용하는 방법을 설명한다.

2.1 Core-Stateless망

Core-Stateless망 구조는 (그림 1)과 같이 나타낼 수 있으며, 흐름단위의 상태 관리를 수행하는 경계노드와 흐름단위 상태 관리를 수행하지 않는 중심노드로 구성된다.

경계노드는 패킷을 흐름단위로 분류하고, 흐름단위로 속도를 예측하여 계산하며, 계산된 정보를 패킷 헤더에 실어 중심노드로 전달한다. 중심노드는 흐름단위의 상태 관리 없이 이레이블 정보를 사용한다. 또한 Core-Stateless망에서는 공정성 보장을 위해 필요한 흐름단위의 버퍼링이나 스케줄링을 하지 않고 각 라우터는 확률적인 폐기알고리즘을 사용하는 FIFO 큐잉을 사용한다. 패킷을 폐기하는 확률은 레이블에 실려 온 정보와 라우터에서 통합 트래픽의 측정에 의한 값에



(그림 1) Core-Stateless Network

의해 결정된다. 패킷 폐기알고리즘도 각 공평 대역 할당 메커니즘의 레이블 정보와 통합 데이터에 따라 다양하게 나타난다[12].

2.2 FNE(Flow Number Estimation) 기법

FNE(Flow Number Estimation) 기법은 문자 그대로 실질적으로 버퍼에서 차지하고 있는 흐름(flow)의 개수를 파악하는 것을 말한다. 망은 지속적으로 데이터를 전송한다고 하더라도 on-off를 가진다. 그러므로 버퍼에 유입된 흐름의 개수를 정확히 산출해 내면, FIFO 큐에서 임계치를 알 수 있으며, 공정하게 대역폭을 할당 할 수도 있다. 기본적으로 이상적인 패킷 폐기를 위해 유입된 양은 식(1)과 같이 링크 용량 C를 넘지 말아야한다.

$$\sum r_i \leq C \tag{1}$$

식(1)에 의해 흐름의 최소값은 $\min(arrival\ rate, threshold\ rate)$ 정의되어 질 수 있다. 만약 n 개의 흐름이 버퍼에 유입될 때 유입된 양이 전체 용량을 넘을 때, 패킷이 폐기된다. 따라서 전체 유입된 도착속도는 $r_1 \leq r_2 \leq \dots \leq r_n$ 으로 정렬 될 수 있다. 따라서 임계치 또는 CL(Cutting Line)은 n+1 중에 하나가 된다. 다시 말해서 $0 < CL \leq r_1, r_1 \leq CL \leq r_2, \dots, r_{n-1} \leq CL \leq r_n, or\ r_n \leq CL < \infty$ 을 의미한다. 따라서 흐름 n+1에서 CL값은 최대값을 식(2)와 같이 얻을 수 있다.

$$\sum_{i=1}^n \min(r_i, CL) = \frac{n \cdot CL + \sum_{i=1}^n r_i - \sum_{i=1}^n |r_i - CL|}{2} \tag{2}$$

따라서, 패킷이 CL을 넘을 시에는 폐기된다.

2.3 흐름속도 측정

2.3.1 흐름속도 측정 메커니즘

유입되는 흐름속도는 얼마나 정확하게 측정되는지의 문제가 고려되어진다. 유입되는 흐름속도를 계산하기 위해서는 FNE(Flow Number Estimation) 기법은 TSW(Time Sliding

Window) 방식을 채택하였다[6]. TSW알고리즘은 흐름속도 측정 기법 중 평균속도를 기반으로, 경계라우터의 서비스 프로파일에 명시된 목표 전송률 초과 여부로 폐기 우선 순위를 결정하여 패킷을 IN이나 OUT으로 마킹하는 알고리즘이다. TSW는 매 패킷이 도착할 때마다 전송속도를 측정하게 되며, 윈도우의 크기에 따라 과거 전송속도에 대한 정보가 현재의 전송속도 계산에 반영되는 정도가 달라지게 된다. 따라서 전송속도 측정부분에서 현재의 전송속도를 과거의 전송속도와 합한 평균으로 버스티한 데이터를 완만하게 만드는 작용을 한다.

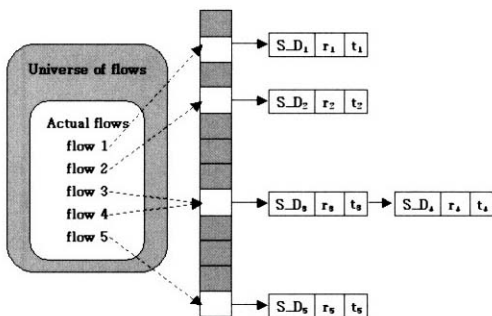
각 흐름의 정보를 기록하기 위해서 m 슬롯을 가진 해쉬 테이블을 사용한다. 이 정보는 각 패킷의 헤더 라벨링을 하기 위해서도 필요하다. 해쉬 테이블에서 해쉬 함수를 통해 슬롯의 위치를 식(3)과 같이 계산하는데, 해쉬 키값은 Src-Dst 주소(Source-Destination address)쌍을 사용한다. 해쉬키 S_{D_i}는 해싱 함수에 의해서 m 슬롯과 매핑된다.

$$h(S_{D_i}) = S_{D_i} \bmod m. \quad (3)$$

여기서 S_{D_i}는 키값이고, m은 슬롯을 의미한다. 해쉬맵으로 흐름단위정보(per-flow information)을 관리하는 것은 흐름단위큐(per-flow queue)보다 메모리사용 효율성이 좋다. 그러나 같은 슬롯에서 매핑하는 과정에서 충돌이 발생할 수 있다. 이러한 문제를 해결하는 방법으로 FNE(Flow Number Estimation)는 chaining 방법을 사용한다. 식(4)는 n개의 흐름이 라우터를 통과 하고 해쉬 테이블이 m개의 슬롯을 가진다고 할 때, 같은 슬롯에 적어도 2개의 flow 맵이 충돌할 확률이다.

$$P_{collision} = 1 - \frac{m! / (m-n)!}{m^n} \quad (4)$$

해쉬 테이블에서 충돌을 방지하기 위해서, (그림 2)에서 보는바와 같이 링크드 리스트를 이용해서 같은 슬롯에 있는 정보를 관리한다. 다시 말해서 흐름단위정보를 슬롯 당 포인터로 관리한다는 의미이다. (그림 2)에서 같이 flow 3과 flow 4는 링크드 리스트에 의해 해쉬되어 같은 슬롯에 위치하고 있다. 이와 같이 Src-Dst 해쉬 키 값과 측정된 흐름속도, 시간과 같은 흐름단위정보만 해쉬의 링크드 리스트에 저장된다.



(그림 2) chaining 방법에 의한 해쉬 테이블에서의 충돌 해결 방법

2.3.2 실제적인 흐름 수 측정

실제적인 흐름 수(Active flows) N_{act}는 큐에서 랜덤하게 선택된 패킷과 유입된 패킷을 비교하여 측정한다. 그러나 패킷 기반 비교측정은 실질적인 네트워크 기반에 적합하지 않다, 왜냐하면 패킷의 길이는 고정적이지 않고 가변적이기 때문이다. 그래서 n 개의 흐름이 라우터를 통과한다고 가정할 때, r_i와 p_i는 흐름 i의 버퍼 용량과 TSW에 의해서 측정된다. 여기서 p_i는 큐에 있는 흐름 i에 속한 패킷의 비율이다. 그리고 N_{act}를 구하기 위해 r_{hit}와 r_{miss}의 값이 식(5)와 식(6)과 같이 산출된다. r_{hit}은 큐에서 랜덤하게 선택된 패킷의 흐름 속도이며 r_{miss}는 같은 흐름에 존재하지 않는 두 패킷의 흐름 속도이다.

$$r_{hit} = \sum_{i=1}^n r_i \cdot P_i \quad (5)$$

$$r_{miss} = \sum_{i=1}^n r_i \cdot (1 - P_i) \quad (6)$$

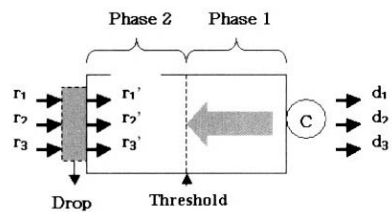
그래서 active flow는 식(7)과 같이 정의되어 진다.

$$N_{act} = \frac{r_{hit} + r_{miss}}{r_{hit}} \quad (7)$$

따라서 FIFO 큐에서 p_i는 r_i에 비례되고, 식(8)과 같이 산출 될 수 있다[6].

$$N_{act} = \frac{n}{C_r^2 + 1} \quad (8)$$

여기서 C_r은 도착속도의 상관관계를 의미하며 δ_r/E[r]과 같다. 공정성을 보장하기 위해 사용되는 fair_rate 지표는 link_capacity/N_{act}이며, 다시 말해서 (그림 3)과 같이 버퍼 임계치가 된다. 이 값은 패킷이 들어 올 때 마다 갱신된다.



(그림 3) FNE의 버퍼관리

간단히 예를 들어 링크 전체 용량은 10M이고, 각 흐름의 흐름속도가 r₁ = 2M, r₂ = 6M, r₃ = 8M이 존재할 때 fair_rate 지표를 4라고 가정한다. 각 흐름에서 예상되는 전송률은 임계치와 비교하여 min(r₁, 4) = 2, min(r₂, 4) = 4, min(r₃, 4) = 4로 나타낼 수 있고 각 임계치를 넘기는 흐름에 있는 패킷의 폐

기 확률은 식(9)와 같다.

$$\begin{aligned}
 p_1 &= 1 - (4 / r_1) = 0 \\
 p_2 &= 1 - (4 / r_2) = 0.33 \\
 p_3 &= 1 - (4 / r_3) = 0.5
 \end{aligned}
 \tag{9}$$

이러한 일련의 과정을 통해 Core-stateless망에서 유출되는 노드와 유입되는 노드를 관리하기 위해, CSFQ보다 더 나은 성능을 위하여 FNE(Flow Number Estimation)를 적용하였으며 이를 CS-FNE라 명명하였다. 그리고 이에 대한 모의실험을 실시하였다.

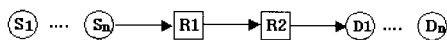
2.4 패킷 라벨링

패킷의 도착속도가 측정되어진 후, 본 연구에서 패킷의 라벨은 해쉬 맵의 슬롯 번호에 따라 할당하였다. 패킷 라벨은 슬롯 m을 가질 때, 확률 1/m을 가지는 $0 \leq label \leq m$ 로 할당하였다[12].

3. 모의실험환경과 성능평가

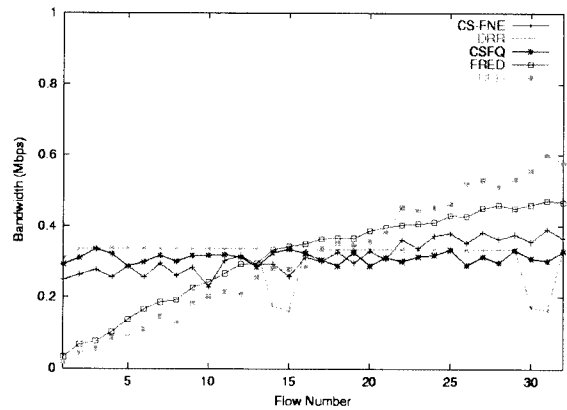
본 연구는 모의실험을 위해 NS-2(Network Simulator)를 사용하였다[10]. 그리고 모의실험에서 사용되는 망 모델을 설명한다. 모의실험 평가의 형평성을 가지기 위해, 기본 망 구성은 CSFQ에서 사용된 단일혼잡망(Simple Congested Link)과 다중혼잡망(Multiple Congested Link)모델을 사용하였다. CS-FNE와 비교되는 알고리즘으로는 DRR(Deficit Round Robin), CSFQ, RED, FRED가 있다. 여기서 WFQ의 변형된 형태인 DRR은 라운드 로빈 방식으로 흐름단위 단위 당 처리되는 큐잉 기법으로 공정한 대역 할당 표현에 있어, 아주 이상적인 공정성을 가지며 하나의 벤치마크로써 존재한다[11].

모의실험 파라미터 설정으로 버퍼 크기는 64000 bytes, 패킷사이즈는 1000 bytes, 링크 지연 1ms이다. CSFQ를 위한 버퍼 임계치는 16000 bytes 이며 CSFQ에서 사용되는 수식에 필요한 상수 K는 100ms, Ka는 100ms로 두었다. RED와 FRED를 위한 최소 임계치는 16000 bytes, 최대 임계치는 32000 bytes로 설정하였다. 그리고 TSW의 win_len(window length)은 0.3초로 하였다. 이러한 파라미터를 가지는 네트워크를 가지고 단일혼잡망과 다중 혼잡망에 대한 모의실험을 하였으며 각 단일혼잡망 다중혼잡망에 대해서 UDP 만이 망을 공유할때와 TCP와 함께 망을 공유할때의 모의실험을 실시하였다.



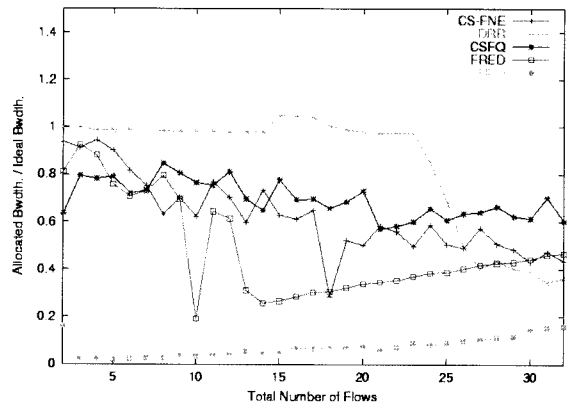
(그림 4) 링크용량 10Mbps, 지연 1ms 인 단일혼잡망

(그림 4)는 모의 실험망으로 단일혼잡망에서 32개의 데이터 흐름을 가정한다. 이망의 첫 번째 실험은 데이터 흐름 전부가 32개의 UDP는 데이터만을 가진다.



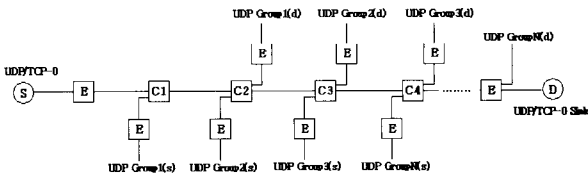
(그림 5) 단일혼잡망, 32개의 UDP 데이터흐름, packet size = 1000 bytes 일 때의 공정 대역폭

(그림 5)에서 나타난 결과는 각 흐름 번호에 대한 평균 처리율이다. 여기서 DRR은 아주 이상적으로 나타났으며, 연구에서 제안한 CS-FNE는 흐름 번호가 증가함에 따라 평균 공정 대역폭도 증가하나 RED와 FRED 보다는 나은 결과를 나타내었으며, CSFQ보다는 평균 공정성에 차이가 있음을 보여주고 있다.



(그림 6) 단일혼잡망, 하나의 TCP 데이터 흐름과 31개의 UDP 데이터흐름, packet size = 1000bytes일 때의 공정 대역폭

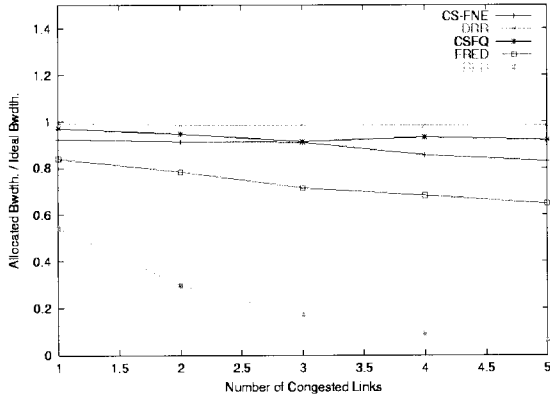
(그림 6)은 하나의 TCP 데이터를 가진 흐름과 31 개의 UDP 데이터 흐름을 가진다고 가정할 때의 모의 실험이다. UDP는 ack 메시지가 없다. 그러나 TCP는 반응적으로 목적지에 도착하면 ack 메시지가 송신지에 보내어진다. 만약 혼잡이 발생되어 패킷이 손실되어지면 slow-start, 빠른 복구 등 TCP의 혼잡 메커니즘이 동작한다. 그래서 병목 구간에서 버스티한 UDP 패킷들이 혼잡이 발생할 때, TCP도 영향을 받게 된다. 다시 말해서 TCP의 혼잡복구 동작이 UDP 데이터 흐름에도 영향을 미친다. 이러한 영향에 대한 실험이다. 제시된 CS-FNE는 공정 대역폭에 대해서 CSFQ보다 진동이 적으나 흐름 번호가 커졌을 때 RED, FRED, DRR 보다는 나은 결과를 나타내고 있다. DRR은 성능이 우수하게 나타났으나, 데이터 흐름 전부가 UDP 일 때의 경우와 다르게 흐름 번호 22부터 성능이 저하되는데, 그 이유는 TCP와 버퍼를 공유하여 영향을 받았기 때문이다.



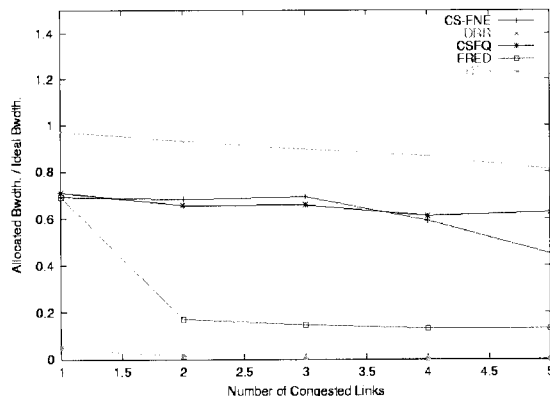
(그림 7) 링크용량 10Mbps, 각 노드간 지연 1ms 인 다중혼잡망

(그림 7)은 GFC(Generic Fairness Configuration)망으로 Stoica의 다중 혼잡망과 같이 구성하였다. 각 링크 10Mbps 용량을 가지며, 지연 1ms로 혼잡 링크에 대한 실험이다. 여기서 S의 소스만 UDP 또는 TCP 데이터로 변경하며, UDP GroupN(s)는 그 다음 노드의 UDP GroupN(d)와 링크가 연결되어 실험하였다.

(그림 8)은 (그림 7)의 다중혼잡망에 대한 모의 실험이다. 여기서 맨 좌측 경계 노드의 소스만 제한된 모든 유입되는 경계 노드는 하나의 그룹에 10개 씩의 UDP소스가 있다고 가정한 결과이다. (그림 8)의 결과에서 DRR은 아주 이상적인 결과 값을 나타내고 있다. 그리고 CS-FNE는 CSFQ와 미세한 차이를 나타내고 있으며, FRED도 전체의 데이터가 UDP 일 경우에 단일망에서의 성능보다 나은 결과를 볼 수 있다. RED는 링크 혼잡의 수가 증가할수록 성능이 저하를 볼 수 있다.



(그림 8) 다중혼잡망, 한 그룹당 10개의 개의 UDP 데이터흐름, S = UDP, packet size = 1000 bytes 일 때의 공정 대역폭



(그림 9) 다중혼잡망, 한 그룹당 10개의 개의 UDP 데이터흐름, S = TCP, packet size = 1000 bytes 일 때의 공정 대역폭

(그림 9)는 (그림 7)의망을 모델로 맨 좌측 경계 노드의 소스를 TCP로 설정하고, 10개씩의 UDP소스를 하나의 그룹으로 하였다. (그림 8)의 결과에서 CS-FNE는 CSFQ보다 혼잡 링크 2와 3.5 사이의 공정 대역폭이 나은 결과를 보이고 있다 그러나 링크 혼잡의 5일 경우 감소하는 것을 볼 수 있다. (그림 8)에서 FRED의 결과는 (그림 9)의 결과와 현격하게 차이를 나타내고 있다.

4. 결 론

FQ, WFQ, FRED 등 많은 흐름단위 스케줄링알고리즘들이 전송률과 지연의 보장을 위해 제안되어 왔었으며, 혼잡제어를 위해 설계되었다. 그러나 흐름단위 처리를 기본으로 하는 알고리즘들은 상태와 버퍼 관리, 패킷 스케줄링을 필요로 하기 때문에, 구현이 복잡하고 데이터 처리에 대한 속도도 늦을 뿐만 아니라, 모든 노드가 패킷의 분류와 흐름단위 혼잡제어를 하기 때문에 고속망에서는 적합하지 않은 문제점이 있다.

본 연구는 이러한 복잡성을 줄이기 위해, FNE의 큐잉 기법을 Core-Stateless망에 적용하였다. 이를 CS-FNE라고 하였고, CSFQ, FRED, RED, DRR과 같은 알고리즘을 함께 평가하였다. 단일 혼잡 공유망에서는 RED, FRED보다 나은 결과를 나타내었으며, 다중혼잡망에서도 RED, FRED 보다 나은 결과를 나타내었다. 그러나 이상적인 공정성에 대한 결과는 산출되지 않았으나, 공정 속도를 지수 분포로 계산하지 않아 구현이 단순화 될 수 있었다. 따라서 고속망에서 계산의 복잡성을 줄이고, 하드웨어적으로 쉽게 구현이 될 수 있다.

향후 과제로는 공정성에 대한 결과를 더욱 향상시킬 필요가 있으며, TCP 데이터 혼잡제어에 대해서도 적용적으로 처리할 수 있도록 개선할 필요가 있다.

참 고 문 헌

- [1] Ion Stoica, Scott Shenker, and Hui Zhang, "Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocation in High Speed Networks", in Proceeding of SIGCOMM'98, Oct., 1997
- [2] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm", J. Internetw. Res. Experience, pp.3-26, Oct., 1990
- [3] Parekh, A. A generalized processor sharing approach to flow control.
- [4] D. Lin and R. Morris, "Dynamics of random early detection", in Proc. ACM SIGCOMM, Cannes, France, Oct., 1997, pp.1427-137.
- [5] Z. Cao, Z. Wang, E. Zegura, "Rainbow fair queueing: fair bandwidth sharing without per-flow state", Proceedings INFOCOM. March, 2000, pp.922-931.
- [6] Jung-Shian Li and Ming-Shiann Leu, "Fair bandwidth share using flow number estimation," Communications,

2002. ICC 2002. IEEE International Conference on, Vol.2, pp.1274-1278, May, 2002.

[7] D .D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service", IEEE Trans (1998), 362-373.

[8] L Zhang, "Virtual clock: a new traffic control algorithm for packet switching networks", in Proc. ACM, SIGCOMM 90, 1990, pp.19-29.

[9] S. Floyd and V. Jacobson, "Random early detection for congestion avoidance", IEEE/ACM Trans. Networking, Vol.1, pp.397-413, July, 1993.

[10] NS simulator, available from <http://www.isi.edu/nsnam/ns>.

[11] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin", IEEE/ACM Trans. Networking, pp.375-385, June, 1996.

[12] 김화숙, 김상하, 김영부, "대역 공평성 보장을 위한 Core-Stateless 기법 연구", 한국통신학회 논문지, Vol.28, No.4C, 2003. 04.

[12] 서경현, 김문경, 육동철, 박승섭, "Core-Stateless망에서의 공정한 대역폭 할당에 관한 연구", 제11회 한국정보과학회 영남지부 학술발표대회, pp.53-59, 2003. 12.



김 문 경

e-mail : geangee@hanmail.net

2003년 부경대학교 멀티미디어공학과(공학사)

2005년 부경대학교 전산교육전공(교육학석사)

관심분야: 네트워크 혼잡제어, 멀티미디어 통신, 코스웨어 설계,



박 승 섭

e-mail : parkss@pknu.ac.kr

1982년 경북대학교 공과대학 전자계산 전공(공학사)

1984년 일본대학 이공학연구과(공학석사)

1984년~1986년 한국통신 연구원

1989년~1990년 일본 동북대학 객원교수

1993년 일본 동북(Tohoku)대학(공학박사)

1998년~1999년 부경대학교 컴퓨터멀티미디어 공학부장

1986년~현재 국립부경대학교 전자컴퓨터정보통신공학부 교수

관심분야: 인터넷 엔지니어링, 초고속통신망, 멀티미디어 통신