

GFR 서비스를 위한 DT-DFBA 버퍼관리 메커니즘

정희원 안상규*, 최승권**, 신승수***, 조용환***

Buffer Management Mechanism Using DT-DFBA Algorithm for GFR Service

Sang-Kyu Ahn*, Seung-Kwon Choi**, Seung-Soo Shin***, Yong-Hwan Cho*** *Regular Members*

요약

FIFO 기반 버퍼관리 알고리즘의 공정성 향상과 단순성을 유지하기 위해 새로운 LBO threshold 결정 메커니즘을 제안하였다. 제안한 알고리즘은 활성화된 각 VC의 가중치 합에 따라 LBO를 차별적으로 계산하여 동적으로 적용한다. 이러한 LBO 설정 메커니즘을 DFBA 알고리즘에 적용한 DT-DFBA(Differential Threshold-Differential Fair Buffer Allocation)을 구현하고 실험을 통하여 성능을 평가하였다.

ABSTRACT

To keep simplicity that needs for GFR service and improve fairness of FIFO based buffer management algorithm, we propose a new LBO threshold decision mechanism. Proposed mechanism uses dynamic threshold that are adjusted according to the sum of active VC's weight. We reformed DT-DFBA(Differential Threshold-Differential Fair Buffer Allocation) algorithm using proposed LBO threshold decision mechanism with existing DFBA algorithm.

1. 서론

ATM Forum은 최근 발표한 Traffic Management Specification 4.1에서 ATM의 여섯 번째 서비스인 GFR(Guaranteed Frame Rate) 서비스의 추가를 확정하였다. GFR 서비스는 원래 UBR^[1]라고 불렀는데 UBR 서비스를 이용하는 TCP/IP 응용들이 직면하게 되는 성능저하 문제를 극복하기 위해 도입되었다. GFR 도입의 배경에 숨어있는 중요한 동기는 UBR의 단순성을 유지하면서 종단시스템에 대한 서비스 기능 강화를 위한 것이다.

GFR 서비스의 구현은 태깅, 버퍼관리, 및 스케줄링 등의 기능 모듈을 이용하여 달성할 수 있다^[2]. 태깅 및 버퍼관리 메커니즘은 각 연결에 대한 FIFO 버퍼의 공간 점유를 제어한다. 스케줄링에 의한 GFR 서비스의 구현 방법도 제안되어 있으며

per-VC queuing을 같이 사용할 경우 서비스 지원이 이루어지게 된다. 그러나 per-VC queuing은 실제 스위치 시스템에 구현하는데 있어서 복잡성과 고비용의 문제가 제기되고 있다. 가능하면 GFR 서비스의 단순성을 유지하면서 최소전송율 보장, 남은 대역의 공평한 할당 및 효율적인 링크 대역의 사용을 보장하는 알고리즘에 대한 연구가 계속되고 있다.

본 논문에서는 효율적인 GFR 서비스의 구현을 위한 FIFO 기반의 버퍼관리 메커니즘을 제안한다. 제안하는 DT-DFBA 알고리즘에서도 기존 버퍼관리 메커니즘과 같이 세 개의 threshold를 사용한다. 그러나 프레임 폐기 기준이 되는 첫 번째 threshold인 LBO를 각 VC에 대하여 일률적으로 적용하지 않고 차별적으로 설정한다. 이는 LBO 설정을 각 VC별로 차별화 함으로서 공정성을 향상시키고자 함이다. LBO의 설정기준은 현재 버퍼용량과는 무관하며, 활성화된 각 VC들의 MCR에 따른 가중치에 기반

* 한국에너지기술연구원 기술확산연구부 정보전산팀(softahn@kier.re.kr), ** (주)애니솔루션

*** 충북대학교 컴퓨터공학과(yhcho@cbucc.chungbuk.ac.kr)

논문번호 : K01216-1020, 접수일자 : 2001년 10월 20일

하여 설정한다.

제II장에서는 GFR 서비스를 위하여 제안된 기존 알고리즘의 개선점 및 문제점 분석을 실시하고 기존 DFBA(Differential Fair Buffer Allocation) 알고리즘의 공평성과 MCR 보장성을 향상시킬 수 있는 LBO 설정 메커니즘 제안한다. 제III장에서는 제안한 메커니즘을 DFBA 알고리즘에 적용한 DT-DFBA 알고리즘에 대하여 버퍼 한계치의 변화에 따른 TCP 소스에 대한 실험을 실시하여 제안한 알고리즘의 성능을 고찰하고 기존 알고리즘과 비교 분석한다.

II. DT-DFBA 알고리즘 제안

1. 기존 알고리즘 분석

GFR 서비스 제공을 위해 제안된 알고리즘을 크게 세가지(Tagging 기반, per-VC accounting and per-VC queueing, per-VC accounting in a FIFO buffer)로 분류하여 분석하였다.

1.1 Tagging 기반 알고리즘

ATM Forum TM 4.1 VI.4.2^[3]에서 제시한 태깅 기반 단일 FIFO 큐 알고리즘(일종의 Double EPD)은 잔여 대역을 각각의 연결에 공평하게 분배하는 어떤 구조도 가지고 있지 않다. 시뮬레이션 결과에 의하면 최소전송율 보장이 TCP 소스 모두에게 보장되지 못함으로 인하여 전체 TCP 성능이 만족스럽지 못한 것으로 나타났다^[4]. 높은 대역의 MCR을 요청한 소스는 요구대역에 훨씬 못 미치는 대역을 할당받았으며 아주 낮은 대역을 요청한 소스는 요구대역보다 훨씬 높은 대역을 할당받았다. 그 이유는 Q(현재 버퍼용량)가 L(Low Threshold)보다 낮을 경우 무조건 모든 셀을 저장하는데 각 연결의 MCR을 참조하는 어떠한 기능도 존재하지 않기 때문이다. Q가 L을 초과할 때 버퍼에 저장되는 셀의 양은 태깅되지 않고 F-GCRA 메커니즘을 통과하는 셀에 의해 결정된다. TCP 트래픽은 버스티한 특성을 가지기 때문에 장기적으로 볼 때 요구한 MCR보다 낮은 전송율을 가지는 TCP 소스에 대해서도 태깅하는 비율이 높다. 결국 TCP 폭주제어 알고리즘이 높은 MCR을 요구하는 TCP 소스에 대해서 적절한 수준 이하의 폭주 윈도우를 설정함에 따라 많은 셀 손실을 유발하게 된다.

Pappu와 Basak은 policing을 사용한 FIFO 기반 메커니즘만으로는 GFR 보장이 불충분하다고 주장

하였다. Per-VC 단일 TCP 연결의 VC를 사용한 실험에서 높은 MCR의 소스가 할당된 대역을 사용하는데 실패한다는 것을 발견하였다^[5]. 그럼에도 불구하고 Bonaventure는^[6]에서 여러개의 TCP 연결들은 단일 VC상에서 라우터에 의해 다중화될 때 MCR 이용이 가능하였다는 것을 찾아내었다. 더욱이 각 라우터에 접속된 소스들은 MBS(Maximum Burst Size)에 의해 더 많은 영향을 받았다. 큰 MBS 즉, MFS의 20배 정도의 MBS를 가진 경우, 소스는 그들의 예약된 대역을 사용하는 것이 가능하였으며 라우터간 지연의 영향은 제한적이었다.^[7]에서 TCP 성능에 있어서 TCP 폭주제어 메커니즘의 영향이 검토되었다. 여러 가지 GFR 포워딩 메커니즘으로 이용하여 실시한 다양한 TCP 버전에 대한 시뮬레이션에서 비록 충분하지는 않았지만 SACK TCP^[8]가 가장 좋은 성능을 보였다. TCP 성능저하를 유발하는 중요한 원인은 이미 알려진 바와 같이 세밀하지 않은 TCP 타이머 granularity를 사용하는 것 외에 FIFO 기반 스위치를 위하여 망 입구에서 트래픽 태깅을 의해 변형된 GCRA를 사용하는 것이다. 그 이유는 TCP는 자신의 전송율을 F-GCRA에 의해 요구되는 트래픽 계약에 적용시키기 어렵다는 것이다. 따라서 만약 FIFO 기반 스위치가 사용된다면, 태깅은 망 내에서 이루어져야 하며 망 접속점에서 이루어져서는 안된다는 것이다. 더욱이 TCP 성능은 다른 GFR 트래픽이 HBO 한 계처럼 링크를 공유함에 의해 특별히 영향을 받는 것은 발견되지 않았다. 작은 값의 LBO가 작은 granularity 타이머를 사용한 TCP에 대해 더 좋은 성능을 보였다. 결국 TCP MSS (Maximum Segment Size)의 값은 TCP 처리율에 영향을 미치지 않으며 모든 TCP 소스에 대해서 MSS는 동일하게 제공된다. TCP 소스들에게 상이한 MSS 값이 제공되면 처리율에 관한 공평성이 심각하게 저해되며 큰 MSS 값이 제공될수록 더 높은 처리율 향상을 기할 수 있다^[7].

Double EPD 알고리즘의 단점을 보완하기 위해 제안된 태깅기반 per-VC 큐잉 알고리즘을 RR 방식으로 스케줄링 할 경우 모든 연결들이 항상 MCR을 달성하지는 못하는 것으로 나타났다^[5]. WRR을 사용할 경우 Q가 L보다 낮으면 모든 대역은 연결들의 MCR에 비례하여 할당되었다. Q가 L을 초과하고 H보다 낮으면 서비스율은 F-GCRA 함수에 의해 결정되었다.^[5]의 시뮬레이션 결과에 의하면 태깅을 기반으로 한 per-VC 큐잉방식은 GFR 서비스

를 지원할 수 있다. 그러나 일부에서는 Q가 대부분의 시간을 L보다 낮은 범위에서 위치하도록 구성된 시뮬레이션의 문제점을 지적¹⁹⁾하기도 하였다.

1.2 Per-VC accounting/queueing 알고리즘

WFQ-like 스케줄링을 하는 per-VC accounting 알고리즘에 대한 성능평가가 이루어졌다^{10, 11)}. 시뮬레이션 결과 GFR.1 연결에 대해서는 단일 FIFO 큐를 사용한 프레임 태깅 기법보다 훨씬 좋은 성능을 보였다. 손실이 발생하지 않는 최소전송율을 효율적으로 보장하였으며 잔여 대역도 각각의 연결의 MCR에 비례하여 할당되었다. 손실이 발생하게 되면 TCP 성능은 나빠지지만 태깅 기반의 FIFO 알고리즘보다는 훨씬 나은 성능을 보였다. 이러한 성능의 저하는 주로 TCP에 의해 이루어지는 재전송 또는 폭주회피 메커니즘에 기인한 것이다. 망의 입구에서 F-GCRA에 의해 프레임 태깅이 이루어지는 GFR.2 연결에 대해서는 GFR.1 보다 낮은 성능을 보여 주었다. 이러한 현상 역시 TCP 트래픽의 버스티한 특성에 기인한 것이다.

CLP 비트 정보를 사용하지 않고 스위칭 요소에서의 셀 적합성 테스트에 기반한 GFS(Global FIFO Scheduling) 알고리즘에 대한 성능 분석이 실시되었다¹⁰⁾. Per-VC accounting과 WFQ-like 스케줄링에 기반한 GFS 알고리즘 시뮬레이션에서 손실이 없는 경우 각각의 연결에 대한 대역보장과 높은 성능의 달성성이 가능하였다. 잔여 대역은 각각의 VC에 동등하게 배분됨도 볼 수 있었다. 셀 손실이 발생할 경우 GFS는 항상 공정한 방법으로 예약된 대역을 보장하지는 못하였으나 WFQ 스케줄링보다는 더 나은 성능을 보여주었다.

1.3 FIFO 버퍼 사용 per-VC accounting 알고리즘

ATM Forum TM 4.1 VI.4.3에 제시된 DFBA 알고리즘에 대한 시뮬레이션이 실시되었다¹¹⁾. 이 실험은 GFR.1 연결에 대한 것으로 다중 TCP/IP 트래픽에 대한 VC 연결에 대해 MCR 보장이 이루어지는 것을 보였다. 평균적인 버퍼 점유율은 MCR 비율을 따르게 되고 FIFO 스케줄링을 할 경우 장기적으로는 MCR 보장이 이루어짐을 나타내었다. 위의 시뮬레이션에서 높은 MCR을 요구하는 VC들이 더 많은 초과 대역을 할당받았으나 실제적으로 MCR 비율에 따르도록 하기 위해서는 파라미터 Zi 값의 선정에 신중을 기하여야 한다.

Goyal 등은 또 ¹²⁾에서 DFBA 기반 메커니즘이

낮은 링크 이용율을 갖는 망에서 SACK TCP의 GFR 서비스 보장을 제공하는지를 조사하였는데, 각각의 연결을 대하여 공평 버퍼용량에 비례하여 달성되는 TCP 성능에 대한 시뮬레이션 결과는 GFR 서비스 보장이 이루어짐을 나타내고 있다.

2. DT-DFBA 알고리즘

GFR 서비스를 구현하기 위하여 다양하게 제안된 알고리즘들에 대한 검토와 이러한 알고리즘들에 대한 성능 분석한 결과, 아직까지는 어떠한 알고리즘도 모든 망 및 사용자 환경하에서 완벽하게 GFR 서비스를 제공하지는 못하고 있었다.

다만 per-VC queueing이나 per-VC scheduling을 사용하는 알고리즘은 일반적 환경에서 GFR 서비스 보장이 이루어지는 것으로 나타났으나 구현에 소요되는 비용이 너무 높아서 큰 단점이 되고 있다. 실험적 분석에 의한 결과¹³⁾에서 보듯이 per-VC 버퍼를 사용하는 FQ 알고리즘은 모든 연결에 대하여 최상의 공평성과 함께 MCR을 보장함을 보여 주었으나 구현 복잡성과 추가적 소요비용으로 인하여 실제 스위치에 구현하기는 어려울 것으로 보인다.

단일 버퍼를 사용하는 FIFO 기반의 알고리즘은 구현비용이 저렴한 반면 버퍼의 출구에서 각 VC들에 대한 구별이 불가능함에 따라 버퍼에 들어온 순서대로 스케줄링할 수밖에 없는 단점을 안고 있었다. 이러한 문제점들을 극복하기 위해 per-VC accounting에 기반한 지능적인 버퍼관리 알고리즘이 필요하게 되었으며 per-VC accounting을 기반으로 하는 여러 가지 알고리즘이 제안되었다. 그러나 동적인 per-VC threshold를 갖는 WBA 알고리즘은 GFR 보장을 이루어내지 못하였으며 DFBA 알고리즘과 TBA(Token based Buffer Allocation) 알고리즘이 GFR 보장에 가장 효율적인 것으로 나타났다.

현재 대부분의 ATM 스위치들이 구현 복잡성을 낮추고 저비용으로 GFR을 지원하기 위해 FIFO 버퍼 구조를 채택하고 있음을 볼 때 여유 대역을 공평하게 할당하고 최소전송율을 보장할 수 있는 좀더 정교한 FIFO 기반 버퍼관리 메커니즘이 필요하다. 따라서 기존의 FIFO 기반 알고리즘의 문제점 분석을 통하여 TCP over GFR 서비스에 더 적합한 FIFO 기반 알고리즘을 유도하는 방법을 검토하였다.

실험적 분석¹³⁾에서 보듯이 FIFO 기반의 알고리즘은 낮은 MCR을 요구하는 연결에 대해서는 공평 처리율보다 더 높은 대역을 할당하는 반면에 높은

MCR을 요구하는 연결에 대해서는 공평대역보다 낮은 대역을 할당하는 경향을 볼 수 있었다. 이러한 결과는 결국 공평성을 저하시키게 되어 링크 사용률이 높거나 각 연결들간의 요구 MCR간의 편차가 클 경우 GFR 서비스 보장에 큰 장애 요인이 된다. 이러한 결과는 대역할당을 좀 더 효과적으로 할 수 있도록 하기 위해 CLP 비트나 셀 폐기의 기준이 되는 threshold를 좀 더 다양하게 활용할 수 있는 버퍼관리 메커니즘이 필요함을 의미한다.

도착한 프레임의 저장을 CLP 비트를 참조하여 결정하는 D-EPD나 DFBA 같은 기존의 알고리즘은 버퍼용량이 좀처럼 LBO 이상으로 올라가지 않는다. 이러한 현상은 CLP1 프레임이 폐기되거나 시간초과가 발생하면 'slow start'와 'congestion avoidance' 메커니즘을 이용하여 TCP 트래픽 제어를 수행되기 때문이다. 즉 버퍼용량이 LBO에 도달했을 때 높은 MCR을 요구하는 소스들을 포함한 거의 모든 TCP 소스들은 하나 또는 그 이상의 CLP1 프레임을 손실하게 되고, 곧이어 폭주 윈도우의 감소를 통하여 전송율을 낮추게 된다. 이 순간 버퍼용량은 LBO threshold 이하로 떨어지게 되며 TCP 소스는 global synchronization^[14] 현상이 나타나게 된다.

폐기 알고리즘은 버퍼용량이 LBO를 넘어야 동작을 시작하게 되나 버퍼용량이 LBO threshold 이하에서 많은 시간을 보내고 있으므로 높은 MCR을 요구하는 TCP 소스들은 공평대역량을 할당받을 수가 없게 되며 이는 결국 공평성 저하를 초래하여 GFR 서비스 보장을 어렵게 하는 것이다.

여기에서는 GFR 서비스 보장을 위하여 기존에 제안된 버퍼관리 알고리즘 중에서 가장 효율적이라고 평가받고 있는 DFBA 알고리즘의 공평성을 개선할 수 있는 메커니즘을 제안한다. 제안한 DT-DFBA(Differential Threshold-Differential Fair Buffer Allocation) 알고리즘은 각각 상이한 MCR를 요구하는 연결에 대하여 동일한 LBO threshold를 적용함으로써 인하여 발생하는 DFBA 알고리즘의 공평성 저하 문제를 개선하기 위하여 LBO threshold를 각각의 연결에 차별적으로 설정하도록 한 알고리즘이다.

기존의 DFBA 알고리즘은 두 개의 static threshold (LBO 및 HBO)와 per-VC dynamic threshold인 $X \cdot Wi/W$ 등 세 개의 threshold를 사용하여 버퍼를 관리한다. DFBA 알고리즘은 static threshold만을 갖는 메커니즘의 경우, 자신에게 설정된 threshold를 초과하면 현재 버퍼의 상태에 관계없이 무조건 폐기되

는 문제점을 개선하였다. 즉 자신의 공평 버퍼점유율보다 더 많이 버퍼를 점유하고 있다고 하더라도 현재 버퍼용량이 여유가 있다면 셀을 받아들이도록 하여 처리율의 향상을 가져오게 되었다. 이러한 성능의 개선은 현재의 버퍼에서 각 VC별로 버퍼를 점유한 정보를 관리함으로써 가능하게 되었다. 아울러 per-VC dynamic threshold를 사용함으로써 특정 VC가 공평 버퍼점유율을 초과하지 않았음에도 불구하고 이미 다른 VC들이 버퍼를 과다하게 버퍼를 점유한 관계로 서비스를 받지 못하는 불공평성도 개선할 수 있었다. 또 $LBO < X < HBO$ 인 경우에 공평 버퍼점유율을 초과하여 VC로 들어오는 셀에 대해서는 비록 해당 셀이 태깅되지 않았더라도 확률적 방법으로 폐기하도록 하여 공평성 향상에 기여하도록 하였다.

제안하는 DT-DFBA 알고리즘에서도 세 개의 threshold를 사용한다. 그러나 프레임 폐기 기준이 되는 첫 번째 threshold인 LBO를 각 VC에 대하여 일률적으로 적용하지 않고 차별적으로 설정한다. 이는 LBO 설정을 각 VC별로 차별화 함으로써 공평성을 향상시키고자 함이다. LBO의 설정기준은 현재 버퍼용량과는 무관하며, 활성화된 각 VC들의 MCR에 따른 가중치에 기반하여 설정한다. 이것은 DFBA 메커니즘의 실험적 분석에서 나타났듯이 높은 MCR을 요구하는 VC가 낮은 MCR을 요구하는 MCR과 동일한 LBO 값을 적용 받음으로 인하여 공평성이 저하되는 문제를 극복하기 위한 것이다. 따라서 제안한 알고리즘에서는 높은 MCR을 요구하는 VC에 대해서는 상대적으로 높은 LBO를 설정함으로써 TCP 폭주 메커니즘의 동작 시점을 늦추도록 간접적으로 제어하도록 하였다. 반면에 낮은 MCR을 요구하는 VC에 대해서는 상대적으로 낮은 LBO를 설정함으로써 TCP 윈도우가 상대적으로 작은 값을 갖도록 간접적으로 제어하여 공평성을 향상시키도록 하였다.

DT-DFBA 알고리즘은 기존 DFBA 알고리즘에서 효율 및 공평성 향상을 위하여 사용하는 각 VC별 버퍼 공평 점유율 검사에 의한 셀 폐기 메커니즘과 CLP0인 셀을 확률적 방법으로 폐기하는 메커니즘을 그대로 사용하게 된다. 제안한 알고리즘의 설명을 위하여 알고리즘에 사용될 변수들을 아래와 같이 정의하였다.

X_i = Number of cells of VC_i in the buffer

X = Total buffer occupancy at any given time

$(X = \sum X_i)$

- Xmax = Buffer capacity
- HBO = High buffer occupancy threshold
- LBO = Standard low buffer occupancy threshold
- LBO_i = Low buffer occupancy threshold of VC_i
- MCR_i = MCR of VC_i
- W_i = Weight of VC_i (W_i = MCR_i / ΣMCR_i)
- W = ΣW_i
- N = Number of active connection
- Z_i = Parameter (0 ≤ Z_i ≤ 1)

각각의 VC_i에 대한 LBO_i의 결정은 여러 가지 방식이 있을 수 있으나 기본적으로는 각 VC들의 MCR 가중치에 기반하여야 한다. 제안 알고리즘에서는 LBO_i의 결정식 (식 1)과 같이 정의하였다.

$$LBO_i = LBO + \left(LBO \cdot \frac{W_i}{W} - \frac{LBO}{N} \right) \quad (1)$$

(식 1)에서 괄호내의 값은 LBO 조정계수를 나타내며 그 중 왼쪽 요소는 각 VC의 MCR 가중치에 따라 결정되는 값이고 오른쪽 요소는 기준이 되는 LBO의 산술평균 값이다. 결국 LBO_i 값은 기준 LBO 값에 조정계수를 합산하여 결정되게 되며 조정계수에 따라 각 VC에 설정되는 LBO_i 값이 기준 LBO 값보다 높거나 낮아지게 되며, 일반적으로 높은 MCR을 갖는 소스는 높게, 낮은 MCR을 갖는 소스는 낮게 조정된다.

제안한 DT-DFBA 알고리즘에서 LBO 설정 메커니즘은 알고리즘 동작 초기에는 기준 LBO 값을 할당하여 사용한다. 그러나 새로운 연결이 설정되거나 기존의 연결이 종료되는 등 활성화된 연결의 수가 변하게 되면 재설정(이) 이루어져야 한다. 따라서 버퍼에 연결된 VC의 수가 일정하면 각 LBO_i 값은 static하나 활성화된 연결의 수가 바뀔 때 따라 dynamic한 특성을 나타내게 된다.

그림 1.에 제안한 DT-DFBA 알고리즘의 operating region을 도시하였다. 제안한 알고리즘의 LBO_i 설정 방식에 따라 기존 DFBA 알고리즘에서는 항상 고정적이었던 LBO 값이 동적으로 변하게 되며 각 VC별로 차별화된다. 기존 DFBA 알고리즘의 operating region ①, ② 및 ③이 영향을 받게 되며 operating region ④는 기존과 동일하게 된다. LBO 값이 증가할수록 폐기 알고리즘이 적용될 region(② 및 ③)의 감소로 인하여 전송효율이 증대될 확률이 높아지게 된다. 아울러 LBO_d 값이 0으로 수렴할수록 셀 폐기 알고리즘이 적용되지 않는 영역(region ①)이 감

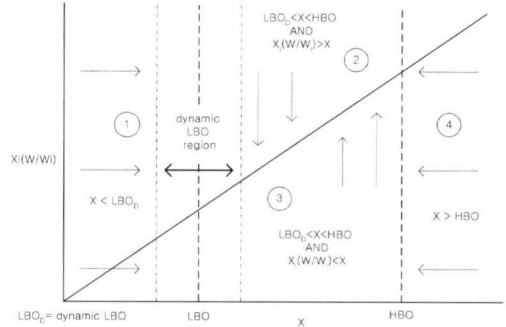


그림 1. DT-DFBA 알고리즘의 operating region

소하게 되어 전송효율이 감소할 확률이 높아지게 된다.

각 VC별로 차별화하여 LBO를 설정함으로써 전송효율의 감소를 최소화하면서 공정성을 향상시킬 수 있는 DT-DFBA 알고리즘의 pseudo code를 그림 2. ~ 5.에 나타내었다. 스위치 초기화 과정에서는 활성화된 연결이 없음을 가정하여 LBO 값을 사용한다. 프레임의 첫 번째 셀이 도착하거나 이미 도착된 프레임의 나머지 셀이 도착하는 알고리즘에서는 새로운 연결의 설정이 감지될 때마다 LBO_i 값을 다시 계산한다. FIFO 버퍼내의 셀을 dequeuing 할 때에는 연결이 종료되는 것을 검사하여 해당 VC의 마지막 셀일 경우에는 LBO_i 값을 다시 계산한다.

```

* Initialization:
X = 0;
N = 0;
W = 0;
For each VC i:
Xi = 0;
Wi = MCRi;
LBOi = LBO;
    
```

그림 2. DT-DFBA 알고리즘의 초기화

```

* When a cell of VC i arrives:
if the cell is the first cell of a frame
if (X < LBOi) or (CLP=0 and Q < HBO
and Xi < X*Wi/W)
accept the cell into the FIFO queue
X++; Xi++; dropi=false;
if (Xi=1)
N++; W = W + Wi;
LBOi = LBO + (LBO * (Wi/W) - (LBO/N))
else
    
```

```

if (CLP=0)&(Q<HBO)&(Xi>X*Wi/W)
if P(drop)<

$$Z_i \left( \alpha \frac{X_i - X * W_i / W}{X(1 - W_i / W)} + (1 - \alpha) \frac{X - L}{H - L} \right)$$

discard the cell; drop=true;
else
accept the cell into the single
FIFO queue
X++; Xi++; dropi=true;
if (Xi=1)
N++; W=W+Wi;

$$LBO_i = LBO + \left( LBO \cdot \frac{W_i}{W} - \frac{LBO}{N} \right)$$

else
discard the cell; dropi=true;
endif
    
```

그림 3. 프레임의 첫 번째 셀 도착시 pseudo code

```

* When a cell of VC i arrives:
if the cell is the subsequent cell of a frame
arrives
if (dropi=true)
discard the cell;
else
if (X<Xmax)
accept the cell into the FIFO queue
X++; Xi++;
if (Xi=1)
N++; W=W+Wi;

$$LBO_i = LBO + \left( LBO \cdot \frac{W_i}{W} - \frac{LBO}{N} \right)$$

else
discard the cell; dropi=true;
endif
    
```

그림 4. 프레임의 연속된 셀 도착시 pseudo code

```

* When a cell with VCi transmitted from the
FIFO queue
X--; Xi--;
if (Xi=0)
N--; W=W-Wi;

$$LBO_i = LBO + \left( LBO \cdot \frac{W_i}{W} - \frac{LBO}{N} \right)$$

    
```

그림 5. FIFO 버퍼내의 셀 dequeue시 pseudo code

III. 성능평가 및 결과분석

효율적인 GFR 서비스 지원을 위하여 제II장에서

제안한 DT-DFBA 알고리즘의 성능평가를 위하여 여러 가지 조건을 설정하여 시뮬레이션을 실시하였다. 제안한 알고리즘의 성능에 대한 비교분석은 주로 DFBA 알고리즘을 대상으로 하였다.

1. 시뮬레이션 환경

1.1 시뮬레이션 도구 및 모듈

시뮬레이션 도구는 NIST ATM/HFC Network Simulator^[15]를 이용하였다. NIST(National Institute of Standards and Technology) ATM 시뮬레이터는 대화식 모델링 환경을 제공하며 UNIX 플랫폼의 X-Windows 시스템을 사용한다. 트래픽을 전송하는 여러 가지 요소들과 B-TE, Switch 및 ATM Application으로 구성되어 있다. NIST 시뮬레이터에는 기본적인 버퍼관리 메커니즘인 EPD(Early Packet Discard), FBA(Fair Buffer Allocation), RED(Random Early Detection) 및 FQ(Fair Queuing) 알고리즘이 구현되어 있다. DFBA, D-EPD, WFBA 등 비교 알고리즘은^[13]에서 구현한 것을 기반으로 하였다. 아울러 여러가지 GFR 서비스 관련 모듈인 셀 순응검사 모듈, QoS 적합성 테스트 모듈, MCR 설정을 위한 인터페이스 모듈등도^[13]에서 추가한 모듈을 이용하였다.

1.2 시뮬레이션 모델 및 트래픽 파라미터

성능평가를 위해 사용된 망의 형태는 그림 6.과 같이 단대단(Peer-to-Peer) 모델로 구성하였다. 이 구성은 다섯 개의 TCP 소스로 구성된다. 트래픽 특성에 관하여 다음과 같은 가정을 하였으며 시뮬레이션에서 사용된 트래픽 파라미터의 값을 표 1.에 나타내었다.

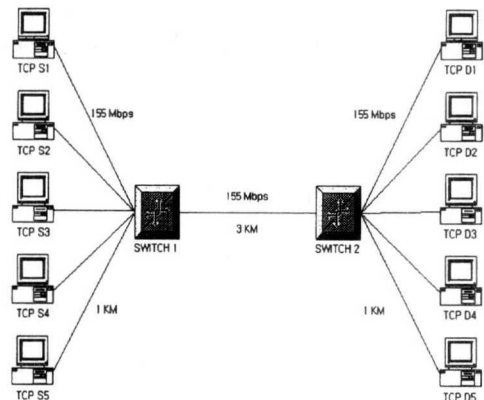


그림 6. Peer-to-Peer Network Simulation Model

표 1. 트래픽 파라미터

ATM Switch	Type	Non-Blocking Output buffer
	Delay to process a cell	4 μ sec
TCP	Windows size	64000 bytes
	Timer granularity	1000 μ sec
	Maximum segment size	9180 bytes
	Mean packet processing time	20 μ sec
	Packet processing time variation	10 μ sec
	Version	Reno
Link	Speed	155 Mbps

- 트래픽은 단방향성을 가지며 소스는 송신만을, 목적지 종단은 수신만을 한다.
- 목적지 종단은 세그먼트의 수신과 동시에 ACK를 한다. 즉 TCP delay ack timer는 설정하지 않는다.
- 모든 TCP 소스는 동시에 출발한다고 가정하며 패킷처리에 소요되는 지연은 표 1.에 나타난 바와 같이 실제의 TCP 구현에서 보다 비교적 적게 설정하였다.
- 소스는 TCP 윈도우가 허용하는 한 세그먼트를 지속적으로 전송한다. 모든 TCP 송신원은 항상 보낼 패킷이 존재하는 무한 소스임을 가정한다.
- 모든 패킷은 시뮬레이터에 의해 최대 패킷크기로 발생한다.

2. 성능평가 기준

2.1 처리율

특정한 버퍼관리 구조를 사용한 TCP over GFR의 성능을 판단하기 위해서는 우선 목적지 TCP 레이어에서 각각의 연결이 연계 되는 처리율(throughput)을 고려하여야 한다. 처리율이란 목적지의 응용으로 전달된 총 바이트 수를 전송에 소요된 시간으로 나눈 값으로 정의할 수 있다. 전송된 바이트에는 재전송 및 손실된 것은 제외된다.

링크상에서 달성 가능한 최대처리율을 C , R bits/sec의 용량으로 네트워크로 전송을 실시하는 TCP 연결의 수를 N , i 번째 TCP 소스의 처리율을 x_i 라 하면 상기의 처리율로부터 효율(efficiency)과 공평성(fairness)이라는 두 가지 성능평가 항목이 유도된다.

2.2 효율

효율은 TCP 연결의 처리율의 합을 하나의 링크 상에서 얻을 수 있는 최대처리율로 나눈 값으로 정의된다.

$$Efficiency = \frac{\sum_{i=1}^N x_i}{C} \quad (2)$$

최대처리율 C 는 데이터에 헤더, 트레일러, 패딩 오버헤드 등이 각각의 레이어에서 추가되므로 인하여 원래의 링크대역보다 적게 된다. 9180바이트의 MSS(Maximum Segment Size)를 갖는 155Mbps 링크상에서의 TCP over GFR 서비스를 예를 들면, 9180바이트의 데이터에 대해서 ATM 레이어는 9180바이트의 데이터와 20바이트의 TCP 헤더, 20바이트의 IP 헤더, 8바이트의 LLC 헤더, 8바이트의 AAL5 트레일러를 받게 된다. 이들은 패딩되어 193개의 ATM 셀로 만들어진다. 따라서, 각 TCP 세그먼트는 ATM 레이어에서는 10229바이트로 구성된다. 이로부터 가능한 최대처리율 C 는 $9180/10229 = 89.7\%$ 로서 이를 대역으로 환산하면 약 139Mbps이다. 아울러 최대처리율은 TCP의 최대 세그먼트에 따라 변하게 되며 MSS가 512바이트인 경우 155Mbps 링크의 최대 처리율은 약 125.2 Mbps로 알려져 있다^[16].

2.3 공평성

망은 높은 처리율을 지향함과 동시에 연결된 각 VC에 대해서 할당된 MCR의 비율에 처리율을 공평하게 할당하여야 한다. 버퍼관리 메커니즘에서 공평성 기준은 equal allocation, weighted allocation, MCR plus equal share, allocation proportional to MCR 등이 있다. MCR 기반의 공평성 기준은 GFR 응용에만 이용이 가능하며 대부분의 UBR 트래픽에 대해서는 equal allocation이 사용된다. 본 논문에서는 MCR plus weighted fair share^[17]를 공평성 척도로 적용한다.

공평성 지수는 TCP 연결상에서 처리율의 차이에 대한 함수로서 (식 3)과 같이 정의되며 0(최소공평성)과 1(최대공평성) 사이에서 분포된다.

공평성 지수(Fairness Index) =

$$\frac{\left(\sum_{i=1}^N \frac{x_i}{e_i}\right)^2}{N \times \sum_{i=1}^N \left(\frac{x_i}{e_i}\right)^2} \quad (3)$$

여기서 x_i 는 i 번째 TCP 연결의 처리율을 나타내며, e_i 는 소스 i 에 대한 공평처리율로 (식 4)로 나타내어진다.

$$e_i = \frac{MCR_i}{\sum_{j=1}^N MCR_j} \times \sum_{j=1}^N x_j \quad (4)$$

3. 결과분석

각 소스의 MCR에 따라 달성될 수 있는 최대처리율의 계산은 다음과 같이 한다. 155Mbps 링크상의 TCP over ATM에서 TCP 최대 세그먼트 크기를 9180바이트로 설정하였으므로 ATM 레이어에서는 이외에도 TCP 헤더, IP 헤더, LLC 헤더 및 AAL5 트레일러가 추가된다. 이것은 193개의 ATM 셀로 패딩된다. 따라서 각각의 TCP 세그먼트는 ATM 레이어에서는 10229바이트가 된다. 이로부터 최대 달성 가능한 처리율은 89.7%(139.04Mbps)로 계산할 수 있다. 본 시뮬레이션에서 사용한 TCP 소스 2의 경우를 예로 들면 $20\text{Mbps} * 0.897 = 17.94\text{Mbps}$ 이다. 각 소스가 달성할 수 있는 이상적인 처리율은 (식 4)에 의해 계산된다. MCR이 20Mbps인 TCP 소스 2의 공평처리율을 계산해보면 다음과 같다.

$$\begin{aligned} Ideal_{MCR=20} &= \frac{MCR_i}{\sum_{j=1}^N MCR_j} \times MaxThroughput \\ &= \frac{20}{75} \times 139.04 \cong 37.07 \end{aligned}$$

3.1 요구대역(MCR)의 합이 링크대역의 50%인 경우

TCP 소스의 MCR을 각각 30, 20, 13, 8 및 4Mbps로 설정하여 총 75Mbps의 대역을 갖도록 하여 출력링크의 50%로 설정하였다. 출력버퍼를 2500셀, HBO를 1500셀로 설정하고 기준 LBO를 각각 1000, 500, 150셀로 설정하였다. 제안한 DT-DFBA 알고리즘에서 LBO 값은 스위치가 초기화될 때만 사용되고 이후에는 제안한 알고리즘에 의해 각 VC 별로 동적으로 변함과 동시에 차별적으로 적용된다.

표 2.에서 제안 알고리즘과 기존 알고리즘의 처리율을 비교하였다. 기준 LBO 값에 관계없이 모두 MCR을 보장하고 있으며 이는 DFBA에서도 동일하게 나타나고 있다.

그림 7. ~ 9.에 LBO에 따른 처리율을 MCR, IDEAL(이상적인 최대처리율) 및 DFBA 알고리즘

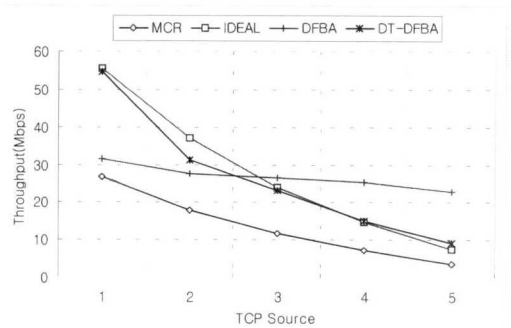


그림 7. 처리율(LBO=1000, $\sum MCR_i$ =링크대역의 50%)

과 비교하여 표시하였다. 모든 LBO 설정값에서 DFBA 알고리즘보다 DT-DFBA 알고리즘이 이상처리율에 근접함을 보여주고 있다. DFBA 알고리즘은 LBO 설정값이 낮아질수록 이상처리율에 근접함을 보였으며 DT-DFBA 알고리즘은 LBO 값이 높아질수록 이상처리율에 근접해나가고 있음을 보여주고 있다. 그러나 이러한 경향들과 전체 처리효율과의 연관성은 나타나지 않았으며 제안 알고리즘의 경우 DFBA 알고리즘 보다 달성율이 전반적으로 1에 가깝도록 분포되었다.

표 2. DT-DFBA 알고리즘의 처리율($\sum MCR_i$ =링크대역의 50%)

LBO	MCR	IDEAL	DFBA (기존)		DT-DFBA (제안)	
			처리율	달성율	처리율	달성율
1000	26.91	55.62	31.60	0.57	54.80	0.99
	17.94	37.08	27.57	0.74	31.16	0.84
	11.66	24.10	26.60	1.10	23.29	0.97
	7.18	14.83	25.44	1.72	15.03	1.01
	3.59	7.42	22.92	3.09	9.29	1.25
500	26.91	55.62	33.62	0.60	45.18	0.81
	17.94	37.08	29.00	0.78	32.04	0.86
	11.66	24.10	26.55	1.10	26.24	1.09
	7.18	14.83	23.87	1.61	18.32	1.24
	3.59	7.42	19.39	2.61	10.94	1.48
150	26.91	55.62	35.92	0.65	39.98	0.72
	17.94	37.08	30.99	0.84	32.27	0.87
	11.66	24.10	26.71	1.11	25.80	1.07
	7.18	14.83	21.50	1.45	19.40	1.31
	3.59	7.42	15.13	2.04	12.05	1.62

* 달성율 = 처리율/IDEAL

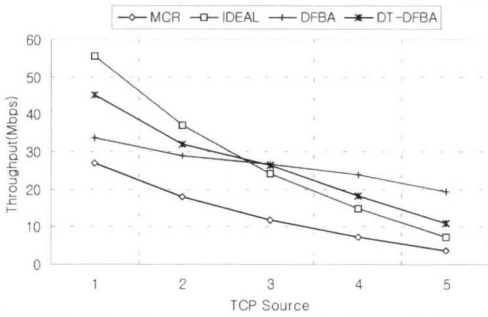


그림 8. 처리율(LBO=500, Σ MCRi=링크대역의 50%)

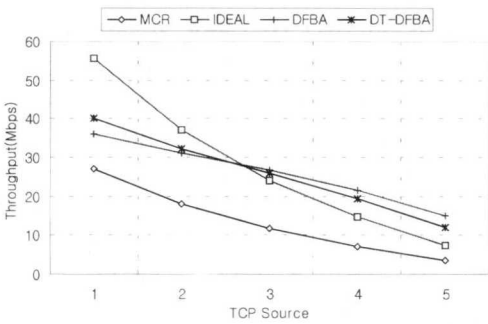


그림 9. 처리율(LBO=150, Σ MCRi=링크대역의 50%)

그림 10.에 제안한 DT-DFBA 알고리즘의 효율을 D-EPD(Double Early Packet Discard), WFBA (Weighted Fair Buffer Allocation), FQ(Fair Queueing) 알고리즘의 효율과 함께 그래프로 나타 내었다. 제안한 알고리즘에서 기존의 DFBA 알고리즘보다 약간의 효율저하가 나타났다. 기존의 알고리즘에서와 마찬가지로 낮은 LBO 값을 설정할수록 효율이 저하되는 경향을 보였다.

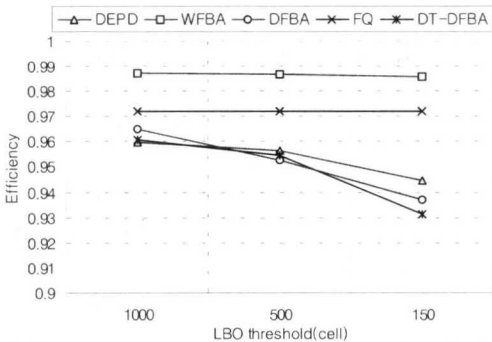


그림 10. 효율(Σ MCRi=링크대역의 50%)

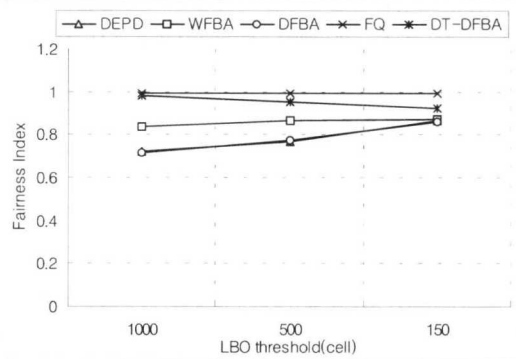


그림 11. 공평성 지수(Σ MCRi=링크대역의 50%)

그림 11.에는 공평성 결과를 나타내었다. 제안한 DT-DFBA 알고리즘은 거의 FQ에 근접하는 정도로 공평성이 향상되었음을 보여주었다. 미세한 차이는 하지만 다른 알고리즘과는 달리 기준 LBO 값이 높을수록 향상된 공평성을 보여주고 있으며 기준 LBO 값에 따른 변화가 심하게 나타나지 않았다.

표 3.에는 MCR 요구대역의 합이 전체대역의 50%일 경우 효율 및 공평성 지수를 기존 알고리즘과 비교하여 나타내었다.

표 3. 효율 및 공평성 지수(Σ MCRi=링크대역의 50%)

Algorithm		DEPD	WFBA	DFBA	DT-DFBA
Efficiency					
LBO	1000	0.9597	0.9870	0.9648	0.9608
	500	0.9562	0.9868	0.9526	0.9547
	150	0.9446	0.9857	0.9369	0.9315
Fairness Index					
LBO	1000	0.7194	0.8361	0.7151	0.9827
	500	0.7658	0.8633	0.7758	0.9527
	150	0.8633	0.8723	0.8589	0.9239

3.2 요구대역(MCR)의 합이 링크대역의 80%인 경우

TCP 소스의 요구 MCR 합이 링크대역의 80%인 경우에 대한 시뮬레이션을 실시하였다. TCP 소스의 MCR을 각각 50, 37, 25, 10 및 2Mbps로 설정하여 총 124Mbps의 대역을 갖도록 하였다. 요구대역(MCR)의 합이 링크대역의 50%로 설정한 시뮬레이션에서와 같이 출력버퍼를 2500셀, HBO를 1500셀, 그리고 기준 LBO 값을 각각 1000, 500, 150셀로 설정하였다.

요구대역(MCR)의 합이 증가함에 따라 기존

표 4. DT-DFBA 알고리즘의 처리율(Σ MCRi=링크대역의 80%)

LBO	MCR	IDEAL	DFBA (기준)		DT-DFBA (제안)	
			처리율	달성율	처리율	달성율
1000	44.85	56.06	33.89	0.60	47.57	0.85
	33.19	41.49	31.85	0.77	37.57	0.91
	22.43	28.03	29.61	1.06	29.37	1.05
	8.97	11.21	21.68	1.93	16.33	1.46
	1.79	2.24	17.97	8.03	4.44	1.98
500	44.85	56.06	36.68	0.65	46.20	0.82
	33.19	41.49	34.35	0.83	36.60	0.88
	22.43	28.03	28.16	1.00	30.68	1.09
	8.97	11.21	21.23	1.89	14.78	1.32
	1.79	2.24	11.37	5.08	4.679	2.09
150	44.85	56.06	38.02	0.68	40.81	0.73
	33.19	41.49	36.97	0.89	37.67	0.91
	22.43	28.03	30.43	1.09	30.78	1.10
	9.97	11.21	17.92	1.60	15.75	1.41
	1.79	2.24	7.47	3.34	4.64	2.07

* 달성율 = 처리율/IDEAL

DFBA 알고리즘은 50Mbps의 MCR을 갖는 연결에 대해서는 MCR 보장이 안되었으나 제안 DT-DFBA 알고리즘에서는 기준 LBO가 150셀인 경우를 제외 하고는 MCR 보장이 가능하였다. 기존 DFBA 알고리즘은 기준 LBO 값에 따라 낮은 MCR 소스에 대한 처리율 변화가 심하였다. 2Mbps의 MCR을 갖는 TCP 소스 5번의 경우는 LBO 설정값이 1000인 경우 요구 MCR보다 795%, 500인 경우 503%, 150인 경우 331%등 평균 543%의 대역을 할당받음으로 인하여 다른 소스에 비해 심각한 처리불공평성을 나타 내었다. 반면에 제안한 DT-DFBA에서는 비교적 일정한 처리율을 보였으며 높은 MCR 소스들은 LBO 값이 커질수록 높은 처리율을 보였다(표 4. 참조).

그림 12. ~ 14.에는 처리율 결과를 도표로 나타 내었다. Σ MCRi가 링크대역의 50%인 경우의 시뮬 레이션 결과와 같이 모든 LBO 설정값에서 DFBA 알고리즘보다 DT-DFBA 알고리즘이 이상처리율에 근접함을 보여주고 있다.

제안한 DT-DFBA 알고리즘의 효율을 비교 알고리즘과 함께 그림 15.에 그래프로 나타내었다. D-EPD 및 WFBA 알고리즘은 Σ MCRi가 링크대역의 50%인 경우와 비교하여 큰 차이가 없었다. 기존 DFBA 및 제안 DT-DFBA 알고리즘은 Σ MCRi가 30% 증가했음에도 불구하고 약간의 효율증가를 나타내었다. 이러한 결과는 모든 TCP 소스에 대하여

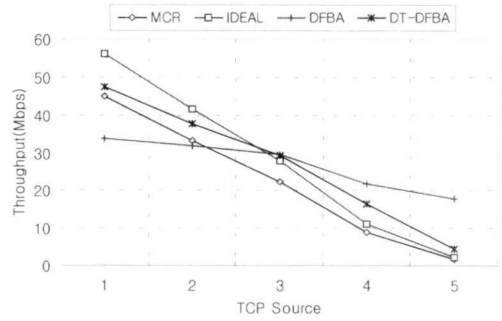


그림 12. 처리율(LBO=1000, Σ MCRi=링크대역의 80%)

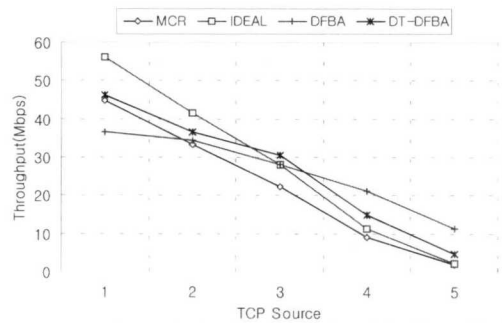


그림 13. 처리율(LBO=500, Σ MCRi=링크대역의 80%)

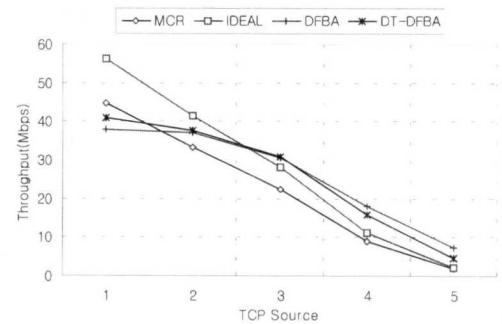


그림 14. 처리율(LBO=150, Σ MCRi=링크대역의 80%)

동일한 평균 패킷처리 시간, 패킷처리 지연시간 및 Timer Granularity의 적용과 함께 스위치 내에서의 셀 처리지연시간등을 동일하게 설정함에 따라서 절 대적인 망의 셀 처리능력이 일정함에 따른 것이다.

모든 알고리즘에서 LBO를 낮게 설정할수록 낮은 효율을 보였으며 낮은 LBO 값의 설정에 따른 효율 저하의 정도는 Σ MCRi가 링크대역의 50%인 경우 보다 심화되었다. 이러한 경향은 LBO 값이 작게 설정될수록 셀 폐기 알고리즘의 동작 빈도가 높아

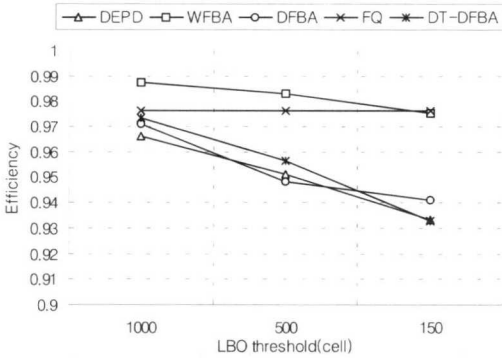


그림 15. 효율(Σ MCRi=링크대역의 80%)

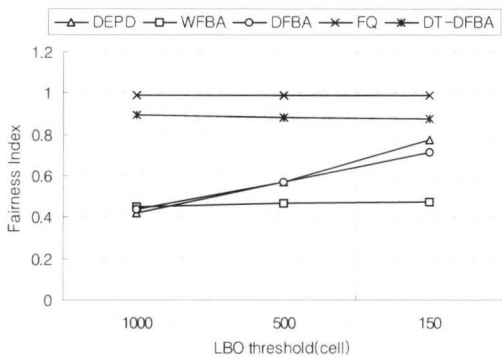


그림 16. 공평성 지수(Σ MCRi=링크대역의 80%)

지는 TCP 폭주제어 메커니즘의 특성에 기인한 것으로 분석된다.

그림 16.에는 공평성 지수를 나타내었다. 비교 알고리즘들은 Σ MCRi가 링크대역의 50%인 경우에 비하여 현격한 공평성 저하를 보이고 있으나 제안한 DT-DFBA 알고리즘은 약간의 저하만을 나타내었다. 아울러 제안한 DT-DFBA 알고리즘은 다른 비교 알고리즘에 비하여 절대적으로 우수한 공평성

표 5. 효율 및 공평성 지수(Σ MCRi=링크대역의 80%)

Algorithm		DEPD	WFBA	DFBA	DT-DFBA
Efficiency					
LBO	1000	0.96	0.98	0.97	0.97
	500	0.95	0.98	0.94	0.95
	150	0.93	0.97	0.94	0.93
Fairness Index					
LBO	1000	0.42	0.45	0.43	0.89
	500	0.57	0.46	0.56	0.88
	150	0.77	0.47	0.71	0.87

지수를 보여주고 있다.

D-EPD 및 DFBA 알고리즘이 LBO 설정값에 따라 공평성 지수의 변화가 심한 반면에 제안 DT-DFBA 알고리즘은 안정된 성능을 나타내었다. 이는 제안 알고리즘이 기준 LBO의 변화에 따른 실제 적용 LBO 값의 변화량 계산을 (식 1)에서의 LBO 조정계수에 따르게 되기 때문이다.

표 5.에는 MCR 요구대역의 합이 전체대역의 80%일 경우의 효율 및 공평성 지수를 기존 알고리즘과 비교하여 나타내었다.

IV. 결론

GFR 서비스는 TCP/IP 응용들이 ATM 망내에서 UBR 서비스 클래스를 이용하면서 격제되는 성능저하 문제를 극복하기 위해 제안되었다. GFR 서비스는 기존의 ATM 서비스 클래스인 UBR 서비스의 단순성을 유지하면서 각 VC에 대해 프레임 기반의 최소전송율을 보장하며 여유 대역이 있는 경우 공평한 할당을 통하여 더욱 높은 효율을 달성할 수 있는 장점을 가진다. GFR 서비스 구현을 위한 여러 가지 알고리즘이 제안되어 있으나 per-VC queuing 방식의 경우 구현복잡성 및 고비용으로 인하여 실제 스위치나 라우터에서의 실현이 어려우며, FIFO 기반의 버퍼관리 메커니즘의 경우 최소전송율 보장 및 공평성 달성이 어려운 실정이다.

본 논문에서는 GFR 서비스가 기반하고 있는 UBR 서비스의 단순성을 유지하고 공평성 및 MCR 보장이 가능한 개선된 DT-DFBA 알고리즘을 제안하였다. 대부분의 FIFO 기반 버퍼관리 메커니즘이 CLP0 셀과 CLP1 셀의 선택적 폐기를 위한 한계치를 각각의 VC가 요구하는 MCR에 대한 고려없이 일률적으로 적용하고 있으며 이러한 버퍼관리 정책은 연결된 VC에 대한 공평성을 심각하게 저하시키게 된다. 제안한 메커니즘은 각각의 연결들이 요구하는 대역에 비례하여 선택적 폐기의 한계값(LBO)을 각 VC별로 차별화하고 활성화된 연결의 가중치에 따라 동적으로 적용한다. 이는 요구 MCR에 비례하는 처리율의 향상을 가져옴으로써 결국 공평성의 향상을 이루어내게 된다. 기존 FIFO 기반 버퍼관리 알고리즘 중 GFR 보장성이 우수한 DFBA 알고리즘의 개선을 위하여 제안한 메커니즘을 사용하였다. 즉 DFBA 알고리즘에서 CLP1 셀의 폐기 및 CLP0 셀의 확률적 폐기를 위한 기준 한계값을 각 VC별로 차별적이며 동적으로 적용하여 처리율 및

공평성을 향상시킬 수 있는 알고리즘을 제안하였다. 제안한 알고리즘의 시뮬레이션을 통해 제안한 버퍼관리 메커니즘에 대한 성능 평가와 기존 알고리즘과의 비교분석을 실시하였다. 실험 결과 제안한 알고리즘이 기존 FIFO 기반 버퍼관리 알고리즘보다 MCR 보장성 및 공평성이 우수함을 보여주었고 TCP의 전송효율은 약간 저하됨을 나타내었다. 그러나 GFR 서비스가 망의 전송효율보다는 각각의 소스들에 대한 MCR 보장과 공평성 향상에 초점을 두고 있음을 보면 제안한 알고리즘은 GFR 서비스를 지원하는 알고리즘으로 활용이 가능할 것이다.

참 고 문 헌

[1] R. Guerin and J. Heinanen, UBR+ Service Category Definition, ATM Forum contribution ATM96-1598, Dec. 1996.

[2] Rohit Goyal, Raj Jain, et al, Design Issues for Providing Minimum Rate Guarantees to the ATM Unspecified Bit Rate Service, Proceedings of ATM98, May 1998.

[3] ATM Forum, ATM Traffic Management Specification, Version 4.1, March 1999.

[4] Bonaventure. O, A Simulation Study of TCP with the GFR Service Category, High Performance Networking for Multimedia Application, 1998.

[5] S. Pappu and D. Basak, TCP over GFR Implementation with Different Service Disciplines: A Simulation Study, ATM Forum contribution ATM97-0310, April 1997.

[6] O. Bonaventure, Providing Bandwidth Guarantees to Internetwork Traffic in ATM Networks, Proceedings of IEEE ATM98 Workshop, Fairfax, USA, May 1998.

[7] O. Bonaventure, A simulation study of TCP with the Proposed GFR Service Category, DAGSTUHL Seminars 9725, High Performance Networks for Multimedia Applications, June 1997.

[8] M. Mathis et al., TCP Selective Acknowledgement Options, RFC2018, Oct. 1996.

[9] K. Spaey, C. Blondia, Buffer Acceptance Schemes for the UBR and GFR ATM Service Categories, Dept. of Telecommunication and Computer Science, University of Antwerp, 2000.

[10] Cerdan, F. and O. Casals, Performance of Different TCP Implementations over the GFR Services category, Special Issue on Multimedia Network and Service Control of the ICON Journal, First Quarter 2000.

[11] Rohit Goyal, Raj Jain, et al., Buffer Management for TCP over the ATM GFR Service, Department of Computer and Information Service, The Ohio State University.

[12] R. Goyal et al., GFR: Providing Rate Guarantees with FIFO Buffers to TCP traffic, ATM Forum contribution ATM97-0831, Sept. 1997.

[13] Yong-Gu JEON, Hong-Shik PARK, Performance Evaluation and Fairness Improvement of TCP over ATM GFR in FIFO-Based Mechanisms, IEICE Trans. Commun.(accepted), 2001.

[14] William Stallings, High-Speed Networks : TCP/IP and ATM Design Principles, Prentice-Hall, 1998.

[15] NIST, "NIST ATM/HFC Network Simulator Ver. 4.1", http://w3.antd.nist.gov/Hsntg/prd_atm-sim.html.

[16] R. Goyal et al., UBR+: Improving Performance of TCP over ATM-UBR Service, Proceedings of ICC '97, June 1997.

[17] Raj Jain. The Art of Computer Systems Performance Analysis. John Wiley & Sons, 1991.

안 상 규(Sang-Kyu Ahn)

정회원



1986년 2월 : 한남대학교 무역학과 졸업
 1996년 2월 : 충북대학교 컴퓨터공학과 석사
 2001년 8월 : 충북대학교 컴퓨터공학과 박사
 1981년 3월~현재 : 한국에너지기술연구원 정보전산팀

<주관심 분야> ATM, MPLS, 통신망

최 승 권(Seung-Kwon Choi)

정회원

한국통신학회 논문지 제25권 제2A호 참조
 현재 : (주)에니솔루션 팀장

신 승 수(Seung-Soo Shin)

정회원

한국통신학회 논문지 제26권 제4A호 참조
 현재 : (주)시그마정보기술 연구소장

조 용 환(Yong-Hwan Cho)

정회원

한국통신학회 논문지 제25권 제9A호 참조
 현재 : 충북대학교 컴퓨터공학과 교수