

고속 통신망을 위한 크레딧 기반 라운드 로빈

정희원 남 홍순*, 김 대영**, 이 형 섭*, 이 형 호*

Credit-Based Round Robin for High Speed Networks

Hong Soon Nam*, Heyung Sub Lee*, Hyeong Ho Lee*, Dea Young Kim** *Regular Members*

요 약

본 논문은 고속통신망에서 고속으로 패킷을 스케줄링하기 위하여 시간 복잡도가 $O(1)$ 인 크레딧 기반 라운드 로빈을 제안한다. 고속통신망을 위한 스케줄러는 연결 수가 많고 연결들의 속도가 다양하며 다양한 크기의 패킷에 대하여 고속으로 스케줄링하여야 한다. 기존의 라운드 로빈 방법은 $O(1)$ 의 시간 복잡도를 갖지만 백로그된 연결에 대하여 한 라운드에서 한번씩 할당된 크기의 패킷을 연속해서 서비스하므로 단시간 공평성과 latency가 저하되고 출력 패킷들이 버스티하게 되는 문제점이 있다. 제안된 방법은 이러한 문제점을 개선하기 위하여 연결 속도에 비례하는 크레딧을 할당하고, 패킷이 도착되면 도착된 패킷의 크기에 해당하는 크레딧을 사용하며 서비스가 완료되면 크레딧을 반환한다. 패킷 도착시 크레딧이 없는 연결은 필요한 크레딧을 기다려서 서비스 받을 수 있다. 본 논문은 제안된 스케줄러의 성능을 분석하였으며, 분석결과 이 방법은 기존의 라운드 로빈 방법에 비하여 단시간 공평성, latency 및 burstiness를 개선할 수 있다.

ABSTRACT

A scheduling scheme for high speed networks requires a low time complexity to schedule packets in a packet transmission time. High speed networks support a number of connections, different rates for each connection and variable packet length. Conventional round robin algorithms have a time complexity of $O(1)$, but their short time fairness, latency and burstiness depend on the quantum of a connection due to serving several packets for a backlogged connection once a round. To improve these properties, we propose in this paper an efficient packet scheduling scheme which is based on the credits of a connection and has a time complexity of $O(1)$. We also analyzed its performance in terms of short time fairness, latency and burstiness. The analysis results show that the proposed scheme can improve the performance compared with traditional round robin schemes. The proposed scheme can be easily utilized in high speed packet networks.

I. 서론

고속통신망에서는 다수의 연결(플로우 혹은 세션)이 제한된 자원을 공유하므로 일시적인 혼잡이 발생될 수 있다. 혼잡 발생시 다수의 연결간 공평성을 제공하고 latency를 제한하기 위하여 스케줄링한다. 스케줄링은 고속통신망을 구성하는 망 요소인 스위치와 라우터의 정합장치 및 스위치 패브릭 등에서 수행되며 네트워크 프로세서 혹은 트래픽 매니저

등의 하드웨어로 구현된다. 고속 통신망에 사용되는 스케줄러는 요구되는 속도를 제공하고 가변길이 패킷 크기를 갖는 다수의 연결에 대하여 구현이 용이하여야 할뿐만 아니라 시간 복잡도가 작아야 한다. 예를 들면 1 Gbps의 속도를 갖는 스위치나 라우터의 포트는 100 바이트의 패킷을 0.8 μ sec 이내에 스케줄링할 수 있어야 한다.

고정 길이 패킷을 서비스하는 ATM(Asynchronous Transfer Mode) 망은 서비스되는 패

* ETRI 네트워크연구소 액세스기술연구부(hsnam, leehs, holee@etri.re.kr)

** 충남대학교 정보통신공학과(dykim@cnu.ac.kr)

※ 본 논문은 2002년 4월 JCCI 학술대회에서 우수논문으로 선정되어 게재 추천된 논문입니다.

킷 수와 전송 속도가 비례하므로 다양한 스케줄러가 사용되고 있다^{[1],[2],[3]}. IP(Internet Protocol)와 같이 가변 길이 패킷을 처리하는 경우 단순히 패킷 수로 스케줄링하면 구현은 용이하지만 연결이 요구하는 속도와 공평성을 보장할 수 없다. 연결이 요구하는 속도와 공평성을 제공하기 위해서는 패킷 수뿐만 아니라 패킷 길이를 고려하여 서비스하여야 한다. 현재 백본을 점유하는 대부분의 패킷 크기는 40 바이트에서 이더넷의 MTU(Maximum Transfer Unit) 크기인 1500 바이트 사이에 분포하고 있지만 일부는 FDDI(Fiber Distributed Data Network)의 MTU 크기인 4000 바이트까지 분포하고 있다^[4]. ATM AAL5^[5] 와 RPR(Resilience Packet Ring)^[6] 프로토콜은 점보 프레임(jumbo frame)을 지원할 수 있다. RPR의 경우 MTU 크기가 9216 바이트로서 작은 패킷인 40 바이트에 비하여 230 배나 크다. 현재 사용되고 있는 스위치나 라우터의 스케줄링 방법은 연결별로 순차적으로 서비스하는 라운드 로빈 방법과 가상 종료 시간의 순서에 따라 서비스하는 공평큐잉 방법이 있다. 라운드 로빈 방법은 구현이 용이하며 시간 복잡도가 작지만 패킷 크기에 따라 단시간 공평성(short time fairness), latency 및 burstiness가 크다는 문제점이 있다. 반면 WFQ(Weighted Fair Queuing)과 같이 가상 종료 시간에 따른 공평큐잉 방법^{[11],[12],[13]}은 각 연결의 선두(HOL: Head of Line) 패킷에 변화가 발생할 때마다 가상 종료 시간(virtual finish time)을 계산하고 계산된 가상 종료 시간이 작은 순서로 패킷을 서비스한다. 이러한 방법들은 시스템 포텐셜을 관리하는 방법에 따라 다양한 알고리즘이 있으나, 계산된 가상 종료 시간에 따라 서비스하기 위하여 연결 수 N 에 대하여 적어도 $O(\log N)$ 의 시간 복잡도가 요구된다.

본 논문은 다수의 연결이 존재하고, 서비스 속도가 다르며, 다양한 크기의 패킷이 존재하는 고속통신망에서 고속으로 스케줄링하기 위한 크레딧 기반 라운드 로빈 (credit-based round robin: CBRR)을 제안한다. 제안된 CBRR은 패킷 크기에 해당하는 크레딧을 사용하여 연결별로 요구하는 속도가 다르며 패킷 크기가 다양한 다수의 연결에 대하여 기존의 라운드 로빈이 가지는 성능을 개선한다. 이하 본 논문은 2장에서 관련 연구 및 문제점을 살펴보고, 3장에서 제안된 CBRR을 설명하고, 4장에서 크레딧 기반 라운드 로빈의 공평성, latency 및 burstiness 특성을 분석하고, 5장에서 결론을 맺는다.

II. 관련 연구

$O(1)$ 의 복잡도를 갖는 고속 패킷 스케줄링 방법은 선입선출 방법과 라운드 로빈 방법이 있다. 라운드 로빈 방법은 단순히 패킷 수만을 고려하여 서비스하는 패킷 기반 라운드 로빈 (PBRR: Packet Based Round Robin)^[7]과 패킷 크기를 고려하여 서비스하는 방법이 있다. PBRR은 구현하기 용이하지만 패킷 크기가 다른 연결들에 대하여 공평성과 연결별로 요구하는 속도를 제공하지 못한다. 패킷 크기를 고려하는 방법은 매 라운드마다 고정 크기로 서비스하는 방법과 현재의 패킷 크기를 고려하여 동적으로 크기를 설정하는 방법이 있다. 고정 크기로 서비스하는 방법으로는 부족 라운드 로빈(Deficit Round Robin: DRR)^[8]과 초과 라운드 로빈(Surplus Round Robin: SRR) 방법이 있다. DRR은 패킷이 도착하면 필요한 퀀텀 크기가 있으면 서비스하고 아니면 다음 라운드에서 퀀텀을 할당받아 서비스하며, SRR은 먼저 서비스하고 다음 라운드에서 초과된 양을 보상하는 방법이다. 한편 가변 크기로 서비스하는 방법은 현재 HOL 크기를 반영하여 서비스하는 동적 부족 라운드 로빈 (Dynamic Deficit Round Robin: DDRR)^[9]과 바로 전 라운드의 서비스 양을 고려하여 서비스하는 탄성 라운드 로빈 (Elastic Round Robin: ERR)^[10] 등의 방법이 있다.

DRR은 연결별로 큐를 가지며 연결 큐는 부족 카운터 DC 와 퀀텀 Q 를 갖는다. 초기화시에 부족 카운터를 0으로 설정하며 퀀텀은 서비스 될 속도에 비례하는 값으로 할당한다. DRR은 패킷이 도착되면 도착된 패킷을 해당 연결 큐에 저장하고 그 연결을 서비스할 차례가 되면 카운터 값을 퀀텀 크기만큼 증가시킨다. 새로운 카운터 값과 연결 큐의 HOL 패킷 크기를 비교하여 HOL 패킷이 카운터 값과 같거나 작으면 서비스하고 서비스된 크기만큼 카운터 값을 감소시킨다. 다시 카운터 값과 새로운 HOL 패킷 크기를 비교하여 패킷 크기가 카운터 값과 같거나 작으면 서비스를 계속하며 그렇지 않으면 다음 연결에 대하여 서비스한다. 이때 연결 i 가 라운드 r 에서 서비스 받은 패킷 크기를 $SQ_i(r)$ 라고 하면, $SQ_i(r)$ 는 식 (1)과 같다.

$$SQ_i(r) = DC_i(r-1) + Q_i - DC_i(r) \tag{1}$$

한편 SRR은 자신의 차례가 되면 HOL 패킷을 서비스하고 DC 를 서비스한 패킷 크기만큼 감한다.

DC가 0 혹은 음수가 될 때까지 계속해서 다음 HOL 패킷을 서비스하고 그 크기만큼 DC를 감소시킨다. DC값이 음수이면 다음 라운드에서 초과된 크기를 보상한다. 이와 같이 DRR과 SRR은 O(1)의 시간 복잡도를 유지하기 위하여 각 연결에 주어지는 퀀텀이 그 연결의 MTU와 같거나 커야 한다. 퀀텀이 크면 한번에 서비스 받을 수 있는 양이 많아지며 그만큼 단시간 공평성과 latency가 저하될 수 있다. 이러한 단점을 개선하기 위하여 ERR과 DRR이 제안되었다.

DDRR과 SRR은 한 라운드에 서비스 될 양을 미리 할당하지 않고 현재 서비스되는 패킷 크기를 고려하여 서비스한다. DRR은 HOL 패킷과 DC 차이가 가장 작은 것을 택하여 이를 퀀텀으로 할당하는 하므로 가장 작은 차이를 찾기 위하여 O(log n)의 복잡도가 요구된다. ERR은 연결별로 초과된 양을 관리하는 초과 카운터 SC와 매 라운드마다 최대 SC를 나타내는 MaxSC를 갖는다. ERR에서 연결 i가 라운드 r에서 서비스 받는 패킷 크기 SQ_i(r)는 다음과 같다.

$$A_i(r) = 1 + MaxSC(r-1) - SC_i(r-1) \quad (2-1)$$

$$SQ_i(r) = A_i(r) + SC_i(r) \quad (2-2)$$

식 (2-1)에서 이전 라운드의 최대 초과 카운터인 MaxSC(r-1)가 SRR의 퀀텀 Q와 동일하다면 ERR과 SRR은 동일한 특성을 갖는다. 그러나 DRR이나 SRR의 퀀텀 Q는 스케줄링되는 연결의 최대 MTU보다 같거나 커야 하므로 ERR은 DRR이나 SRR에 비하여 공평성과 latency를 개선할 수 있다. 그렇지만 연결별로 가중치 W_i를 갖는 경우에는 다음과 같이 MaxSC를 찾기 위하여 SC_i를 W_i로 나누어야 하며, 또 퀀텀을 할당할 때에는 반대로 W_i를 곱해야 한다.

$$MaxSC = Max MaxSC, SC_i / W_i, \forall i \quad (3-1)$$

$$SQ_i(r) = W_i(1 + MaxSC(r-1) - SC_i(r-1) + SC(r)) \quad (3-2)$$

이와 같이 ERR은 연결별로 가중치 W를 제공하기 위하여 곱셈과 나눗셈의 연산이 요구된다. 또한 이러한 라운드 로빈 방법은 백로그된 연결에 대하여 한 라운드에 한번씩 서비스하며 한번 서비스하면 식(1) 혹은 (2)와 같이 해당 연결에 대하여 몇 개의 패킷을 연속적으로 서비스한다. 몇 개의 패

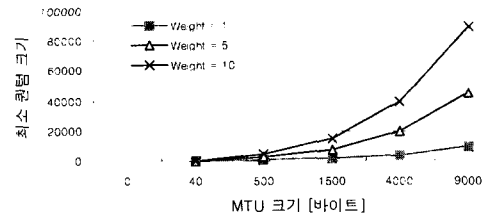


그림 1. 가중치에 따른 최소 퀀텀 크기

킷들을 연속적으로 서비스하면 트래픽 특성이 버스트하게 되며 공평성과 latency가 저하되는 문제점이 있다. 그림 1은 MTU 크기와 가중치를 고려할 때 O(1)의 복잡도를 갖기 위해 요구되는 퀀텀 크기를 나타낸다. MTU가 4,000 바이트이고 가중치가 10이면 최소 퀀텀은 40,000 바이트가 되므로 한번 서비스 받게 되면 40,000 바이트를 연속해서 서비스 받아 트래픽 특성이 버스트해지는 문제점이 있다.

III. 크레딧 기반 라운드 로빈

DRR과 ERR과 같이 기존의 라운드 로빈은 한 라운드에 한번씩 서비스하므로 트래픽 특성이 버스트해진다. 이러한 문제점을 개선하기 위하여 CBRR이 제안되었다. CBRR은 O(1)의 복잡도를 가지면서 기존의 라운드 로빈의 단점인 퀀텀보다 작은 패킷에 대한 burstiness를 완화하기 위하여 그림2와 같이 패킷 풀, 연결 관리부, 연결 테이블 및 토큰 큐로 구성된다. 그림에서 패킷 풀은 패킷을 저장하는 곳으로서 버퍼로 구성되며 연결별 큐를 갖는다. 토큰 큐는 서비스 될 연결 정보를 저장하는 큐로서 선입선출(FIFO)로 서비스된다. 연결 테이블은 연결별로 퀀텀 크기 Q, 가용 크레딧 AC, 요구 크레딧 RC, HOL 패킷 크기 PS, 백로그된 패킷 크기 BS 등 연결에 필요한 정보를 저장한다. 연결 관리부는 패킷이 도착하면 패킷 풀에 저장하고 스케줄링하여

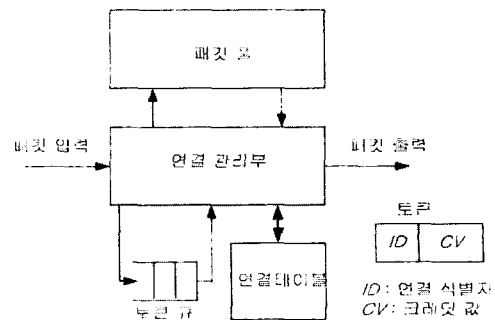


그림 2. CBRR 구성도

출력하는 과정을 관리한다. 연결 관리부는 스케줄러로 입력되는 패킷을 먼저 패킷 풀에 저장하고 해당 연결의 연결 식별자 ID와 서비스 받을 크레딧 값 (Credit value: CV)을 토큰에 실어 해당 토큰 $T<CV, ID>$ 을 토큰 큐에 저장한다.

CBRR은 그림 3과 같이 초기화 과정, 패킷 도착 과정과 출력 과정으로 구성된다. 초기화 과정은 스케줄러의 시작시 수행되며 가용 크레딧, 요구 크레딧 및 저장된 패킷 크기를 0으로 설정한다. 패킷 도착 과정은 패킷이 도착하면 시작되며 도착한 패킷을 패킷 풀에 저장하고 가용 크레딧과 요구 크레딧 크기를 확인한다. 확인 결과 $RC=0$ 이고 $AC_i > SP_i^j$ 이면 $CV=SP_i^j$ 및 식 (4-1)과 같이 AC_i 를 갱신하고 토큰 $T_i<ID, CV>$ 을 토큰 큐에 저장한다. $RC=0$ 이고 $AC_i > SP_i^j$ 아니면 $RC_i = SP_i^j$ 로 설정하고 종료하며 RC 가 0이 아니면 종료한다.

$$AC_i = AC_i - CV \quad (4-1)$$

패킷 출력 과정은 토큰 큐에 저장된 HOL 토큰 $T_i<ID, CV>$ 의 연결 식별자와 크레딧 크기를 참조하여 서비스 한다. 먼저 해당 연결의 HOL 패킷을 서비스하고 식 (4-2)와 같이 AC_i 를 갱신하고 CV 를 패킷 크기만큼 감소시키고 CV 가 0으로 될 때까지 반복하여 HOL 패킷을 서비스한다.

$$AC_i = AC_i + CV \quad (4-2)$$

AC_i 가 RC_i 보다 크면 CV 를 RC_i 만큼 증가시키고 다음 패킷의 크기를 RC_i 로 갱신하고 AC_i 가 RC_i 보다 적거나 다음 대기 패킷이 없을 때까지 반복한다. CV 가 양수이면 토큰 $T_i<ID, CV>$ 를 토큰 큐의 마지막에 저장하며 아니면 종료한다. 만약 Q_i 를 M_i 보다 작게 설정하는 경우에는 시간 복잡도가 $O(1)$ 을 만족하지 못하지만 $AC_i \geq Q_i$ 이고 $RC_i > AC_i$ 이면 $CV=AC_i+Q_i$ 로 설정하여 $CV \geq SP_i$ 가 될 때까지 몇 번의 라운드를 반복하여 서비스할 수 있다.

이와 같이 CBRR은 연결별로 가용 크레딧을 사용하여 속도를 제어하고 가용 크레딧의 상태에 따라 도착하는 패킷을 서비스함으로써 latency와 burstiness를 개선한다

그림 4는 DRR, WFQ, CBRR에 대한 누적 서비스 크기를 나타낸다. 그림 4 (a)는 패킷 도착 과정을 나타낸 것으로 패킷의 마지막 바이트가 도착하면 패킷이 도착한 것으로, 서비스의 경우 패킷의 마지막 바이트가 서비스되면 패킷이 서비스되었다고

```

// Definition
// Q : quantum of a connection
// AC : available credit
// RC : request credit
// PS : HOL packet size
// BS : backlog size

//Initialization process
Set  $Q_i$  for each connection
 $AC_i = Q_i$ ;  $RC_i = 0$ ;  $BS_i = 0$ ;

//Packet arrival process ( $P_i^j$ )
Store  $P_i^j$  in packet queue
 $SP_i^j = P_i^j$  size;  $BS_i = BS_i + SP_i^j$ ;
If  $RC_i = 0$  then {
    If  $AC_i > SP_i^j$  then {
         $CV=SP_i^j$ ;  $AC_i=AC_i-CV$ ;  $ID=i$ ;
        Store Token  $T_i<ID, CV>$ ;
    }
    else {
         $RC_i = SP_i^j$ ;
    }
}

//Packet departure process
//Read HOL token  $T_i<ID, CV>$  from token queue
// $SP_i =$  HOL packet size of connection  $i$ ;
 $AC_i=AC_i+CV$ ;  $BS_i =BS_i-CV$ ;
While  $CV \geq SP_i$  then {
    // Transmit HOL packet of connection  $i$ ;
     $CV=CV-SP_i$ ;
}

While  $AC_i > RC_i$  {
     $CV=CV+RC_i$ ;  $AC_i=AC_i-RC_i$ ;
}
//update next  $RC_i$ =next packet size or 0 (empty)
}
// if  $AC_i \geq Q_i$  &&  $RC_i > AC_i$  then  $CV=AC_i+Q_i$ ;
// for  $Q_i < M_i$ 
if  $CV > 0$  then {
    Store token  $T_i<ID, CV>$ ;
}
    
```

그림 3. CBRR Pseudo-Code

가정하고 임펄스로 표기한다. 그림에서 패킷은 크기가 1인 고정 크기 패킷으로 가정하였고, 연결1, 2, 3은 각각 4:2:1의 가중치를 갖는다고 가정하였다. 그림 4 (b)의 DRR은 시간 2와 9에서 4 패킷씩 버스트하게 서비스하며 그 동안 연결 2, 3은 서비스 받지 못한다. 그림 4 (c)는 가상 종료시간을 기준으로 서비스하는 WFQ의 패킷 서비스를 나타낸 것으로 서비스 크기가 선형적으로 증가한다. 그림 4 (d)의 CBRR은 시간 2, 3, 4에서 연결 1, 2, 3이 하나의 패킷씩 서비스 받고, 연결1의 burstiness는 시간 7에서 3으로 DRR 보다 완화되어 WFQ와 유사하게 서비스한다.

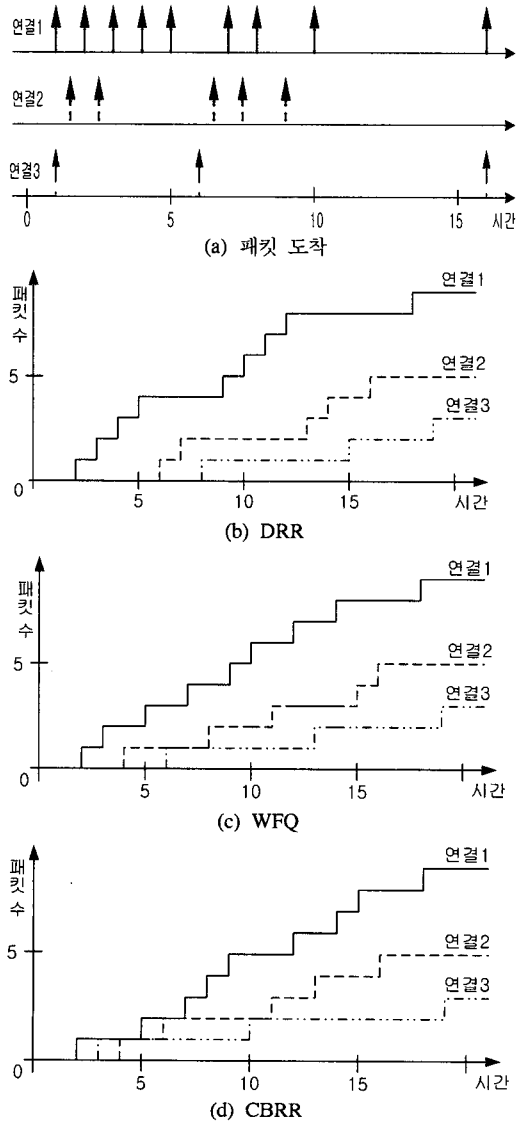


그림 4. 스케줄러별 누적 서비스 (DRR, WFQ, CBRR)

V. CBRR 성능 분석

본 장에서는 CBRR의 특성을 살펴보기 위하여 라운드 로빈 서비스로 발생하는 burstiness를 라운드 로빈 burstiness로 정의하고, 시간 복잡도, 공평성, latency 및 라운드 로빈 burstiness를 분석한다.

1. 시간 복잡도

CBRR은 그림 3과 같이 초기화 과정, 패킷 도착 과정과 패킷 출력과정으로 구성되며, 가산기, 감산기 및 비교기로 구현된다. 패킷 도착과정은 도착한 패킷의 가용 크레딧이 도착한 패킷보다 큰지를 판단하여 크면 토큰을 생성하여 토큰 큐에 저장하고 아니면 종료하므로 $O(1)$ 의 복잡도를 갖는다. 반면 패킷 출력과정은 토큰 큐의 HOL 토큰이 지정하는 연결의 패킷을 서비스하고 크레딧 값에 따라 다음 HOL 패킷을 서비스할 것인가를 판단한다. 이에 따른 파라미터의 갱신은 연결 수와 무관하게 처리되므로 $O(1)$ 의 복잡도를 갖는다. 따라서 패킷 도착과정과 출력과정이 모두 $O(1)$ 의 복잡도를 갖는다.

2. 공평성

참고 문헌[2]는 연속적으로 백로그 된 임의의 2개의 연결 i, j 에 대하여 구간 (t_1, t_2) 에서 받은 서비스 양 $W_i(t_1, t_2), W_j(t_1, t_2)$ 를 예약 속도로 정규화된 차의 최대치를 스케줄러의 공평성 FN 이라고 하고, FN 을 다음과 같이 정의하고 있다.

$$\left| \frac{W_i(t_1, t_2)}{\rho_i} - \frac{W_j(t_1, t_2)}{\rho_j} \right| \leq FN \quad (5)$$

CBRR에서 연결 i 의 가중치를 Q_i , 연결 i 의 최대 패킷 크기를 M_i , 백로그 시작점을 τ , k 라운드에서 연결 i 의 마지막 패킷이 서비스가 종료되는 시간을 t_k 라고 하면 k 라운드 동안 연결 i 가 받는 서비스 크기 $W_i(\tau, t_k)$ 는 다음과 같다.

$$kQ_i - M_i \leq W_i(\tau, t_k) = kQ_i - AC_i(k) \leq kQ_i \quad (6)$$

또 백로그 기간 중 임의의 라운드 시작 시간 t_0 부터 k 라운드 동안 연결 i, j 가 받는 서비스 크기를 $W_i(t_0, t_k), W_j(t_0, t_k)$ 라고 하면

$$kQ_i - M_i \leq W_i(t_0, t_k) \leq kQ_i + M_i \quad (7-1)$$

$$kQ_j - M_j \leq W_j(t_0, t_k) \leq kQ_j + M_j \quad (7-2)$$

으로 된다. 이를 식 (5)에 대입하고, t_1 부터 임의의 시간 t_2 까지 서비스 받는 크기 $W_i(t_1, t_2)$, $W_j(t_1, t_2)$ 를 각각의 속도인 ρ_i , ρ_j 로 정규화하면

$$\left| \frac{W_i(t_0, t_2)}{\rho_i} - \frac{W_j(t_0, t_2)}{\rho_j} \right| \leq \frac{M_i}{\rho_i} + \frac{M_j}{\rho_j} + \text{Max} \left[\frac{Q_i}{\rho_i}, \frac{Q_j}{\rho_j} \right] \leq \frac{n(2M+Q)}{r} \quad (8)$$

으로 된다. 여기서, $\rho_i = \frac{Q_i}{F}$ 이며 $F = \sum_{i=1}^n Q_i$ 이다. M 과 Q 는 스케줄러의 모든 연결에 대하여 최대 패킷 크기와 최대 r 중치를 나타낸다. 식 (8)의 $\text{Max} \left[\frac{Q_i}{\rho_i}, \frac{Q_j}{\rho_j} \right]$ 은 서비스 받은 구간이 라운드의 정수 배가 아닌 경우에 추가된다. 이는 서비스 받은 구간이 라운드의 정수 배인 경우에는 0가 되어 SCFQ의 공평성과 동일하며 정수 배가 아닌 경우에는 $\text{Max} [Q_i/\rho_i, Q_j/\rho_j]$ 만큼 증가된다.

3. Latency

본 논문에서는 임의의 연결의 첫 번째 패킷이 도착하여 서비스가 완료되기까지 소요되는 시간을 시작 지연 $D_{start-up}$ 이라고 정의하고, 참고문헌 [2], [10]과 같이 latency를 시작 지연 시간의 최대값으로 정의한다.

시간 τ 에 패킷 크기가 m_i 인 연결 i 의 첫 번째 패킷이 도착하였다고 가정하고 패킷 도착시간의 AC 를 $AC(\tau)$ 라고 하면 시작 지연 $D_{start-up}$ 은 식 (9)과 같다.

$$D_{start-up} \leq \frac{1}{r} \sum_{j=1, j \neq i}^n (Q_j - A_j(\tau)) + \frac{m_i}{r} \quad (9)$$

latency L 은 정의와 같이 $L \geq D_{start-up}$ 이며, 식 (10)과 같이 가용 크레딧 $AC(\tau)$ 가 0이고 도착한 패킷

크기가 가장 큰 M 인 경우 최대가 된다.

$$L \leq \frac{1}{r} \sum_{j=1, j \neq i}^n Q_j + \frac{M_i}{r} \leq \frac{(n-1)Q}{r} + \frac{M}{r} \quad (10)$$

즉, 시간 τ 에 유휴 연결의 첫 번째 패킷이 도착하면 해당 토큰이 토큰 큐의 제일 마지막에 저장되므로 자신의 앞에 있는 토큰들이 전부 서비스 받은 후에 자신이 서비스 받게 된다. 임의의 연결 j 의 토큰 큐에 저장된 크레딧은 $Q_j - AC_j(\tau)$ 으로 되어 전체 토큰 큐에 대기 중인 크레딧은 이들의 합이 된다.

4. Burstiness

DRR은 자신의 차례가 되면 부족 카운터에 해당하는 크기의 패킷들을 서비스하므로 burstiness가 증가될 수 있다. 본 논문에서는 이러한 burstiness를 나타내는 파라미터로서 라운드 로빈 burstiness를 정의한다. 라운드 로빈 burstiness는 패킷 크기와 속도가 같은 임의의 두 연결 i, j 에서 도착 과정이 $A_i^k \leq A_j^k \leq A_i^{k+1} \leq A_j^{k+1} \leq \dots$ 같이 서로 교대로 패킷이 도착하는 경우 패킷 출력 과정이 $O_i^k \leq O_j^{k+1} \leq O_j^{k+2} \leq \dots \leq O_j^{k+n} \leq O_i^{k+1} \leq O_i^{k+2} \leq \dots \leq O_i^{k+m} \leq O_j^{k+n+1}$ 같이 n 혹은 m 개의 패킷이 연속으로 출력되는 경우 $\max\{m, n\}$ 을 라운드 로빈 버스트니스라고 정의한다.

DRR과 ERR은 연결 i 의 킨텀 크기를 Q_i , 패킷 크기를 m_i 이라고 하면 최대 burstiness는 $\lceil Q_i / m_i \rceil$ 로 증대될 수 있다. 여기서 $\lceil x \rceil$ 는 x 보다 작지 않은 최소 정수를 나타낸다. 한편 CBRR에서는 가용 크레딧이 있는 연결들의 패킷들이 도착하는 경우 도착하는 순서에 따라 패킷을 서비스한다. 단 동일 연결의 직전의 패킷이 큰 패킷이라면 그 크기만큼

표 1. 스케줄링 알고리즘 비교

버스트하게 서비스 될 수 있다.

표1에 스케줄러의 특성을 비교하였다. CBRR은 동일한 퀀텀 Q 를 가지는 경우 공정성은 DRR과 동일하지만 latency 및 라운드 로빈 버스트니스에서 우수한 특성을 갖는다. 기존의 라운드 로빈은 한 라운드에 한번씩 할당된 퀀텀 크기의 패킷을 버스트하게 서비스하기 때문에 발생하지만 CBRR은 퀀텀 이내의 패킷들은 도착된 순서를 고려하여 서비스하기 때문이다.

V. 결론

본 논문은 고속 스케줄러용 크레딧 기반 패킷 스케줄러(CBRR)의 알고리즘 및 구조를 제시하였다. 고속 통신망은 많은 연결에 대하여 가변 길이 패킷을 고속으로 처리해야 하므로 시간 복잡도가 낮은 스케줄러를 사용하여야 한다. 종래의 라운드 로빈 알고리즘들은 $O(1)$ 의 시간 복잡도를 갖지만 한 라운드에 한번씩 서비스하므로 공정성, latency 및 burstiness 등의 특성이 한 라운드에 서비스 될 양인 퀀텀 크기에 비례하였다.

망이 고속화되면서 큰 패킷은 작은 패킷에 비하여 수백 배나 크고 연결별로 다양한 대역폭을 요구하므로 한 라운드에 서비스해야 되는 양인 퀀텀이 증가하게 된다. 기존의 라운드 로빈 방법은 퀀텀 크기가 증가되면 latency와 burstiness가 저하되었다. 이러한 문제점을 개선하기 위하여 CBRR은 각각의 연결에 할당된 퀀텀보다 작은 패킷이 도착하는 경우 가용 크레딧의 상태에 따라 도착하는 순서로 서비스한다. 이러한 방법은 기존의 라운드 로빈에 비하여 latency 및 burstiness를 개선할 수 있다.

성능 분석과 CBRR은 SCFQ에 비하여 공정성 및 latency가 저하되지만 복잡도가 낮고, DRR이나 ERR에 비하여 latency가 양호하고 burstiness를 발생하지 않는 장점이 있다. 또, 시간 복잡도가 낮고 가산기, 감산기 및 비교기로서 구현되므로 고속 통신망에 사용되는 스위치나 라우터에 용이하게 활용될 수 있다.

참고 문헌

[1] D. Stidialis, A. Varma, "Efficient Fair Queueing Algorithms for Packet-Switched Networks," *IEEE/ACM Trans. on Networking*, Vol. 6, No. 2, pp. 175-185, Apr. 1998.

[2] Jong-Seon Kim and Lee, D.C., "Weighted round robin packet scheduler using relative service share," *Proc. MILCOM2001*, pp. 988-992, 2001.

[3] G. Chuanxiong, "SRR: An $O(1)$ Time Complexity Packet Scheduler for Flows in Multi-Service Packet Networks," *Proc. ACM SIGCOMM'01*, pp. 211-222, Aug. 2001.

[4] Phil Dykstra, "Gigabit Ethernet Jumbo Frames," <http://sd.wareonearth.com/~phil/jumbo.html>

[5] IETF RFC1577, "Classical IP and ARP over ATM," IETF, Jan. 1994.

[6] IEEE Draft P802.17/D0.3, "Resilient Packet Ring access method and physical layer Specifications," Jun. 10, 2002.

[7] J. Nagle, "On Packet Switches with Infinite Storage," *IEEE/ACM Trans. on Comm.*, vol. 4, pp.375-385, May 1996.

[8] M. Shreedhar and G. Varghese, "Efficient Fair Queueing using Deficit Round Robin," *IEEE/ACM Trans. on Networking*, vol. 4, pp.375-385, May 1996.

[9] K. Yamakoshi, et al., "Dynamic Deficit Round-robin scheduling scheme for variable-length packets," *Electron. Lett.*, Vol. 38, pp. 148- 149, Jan. 2002.

[10] S.S.Kenhere, H. Sethu, and A.B. Parekh, "Fair and Efficient Packet Scheduling Using Elastic Round Robin," *IEEE/ACM Trans. on Parallel and Distributed Systems*, Vol. 13, No. 3, Mar. 2002.

[11] A. K. Parekh, R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Trans. on Networking*, Vol. 1, No. 3, Jun. 1993.

[12] S. J. Golestani, "A Self-Clocked Fair Queueing Scheme for Broadband Applications," *Proc. INFOCOM94*, pp. 636-646, Apr. 1994.

[13] A. Demers, S. Keshav, and S. Shenker, "Design and Analysis of a Fair Queueing Algorithm," *Proc. ACM SIGCOMM*, pp. 1- 12, Sept. 1989.

남 흥 순(Hong Soon Nam) 정회원



1987년 2월 : 서울시립대학교
전자공학과 학사
2001년 2월 : 충남대학교
정보통신공학과 석사
2001년 3월~현재 : 충남대학교
정보통신공학과 박사과정

1987년 2월~현재 : 한국전자통신연구원 메트로이더
넷팀 책임연구원

<주관심 분야> 트래픽제어, IP 멀티캐스팅

김 대 영(Dae Young Kim) 정회원

한국통신학회 논문지 26권 1B호 참조
충남대학교 정보통신공학과 교수

이 형 섭(Heyung Sub Lee) 정회원



1985년 2월 : 충남대학교 전자
공학과 학사
1994년 8월 : 충남대학교 전자
공학과 석사
2002년 8월 : 충남대학교 전자
공학과 공학박사

1990년 7월~현재 : 한국전자통신연구원 메트로이더
넷팀장

<주관심 분야> 고속LAN 및 라우터 기술, 이더넷
기술

이 형 호(Hyeong Ho Lee) 정회원



1977년 2월 : 서울대학교 공업
교육과 전자전공 학사
1979년 2월 : 한국과학기술원
전기 및 전자공학과 석사
1983년 8월 : 한국과학기술원
전기 및 전자공학과
공학박사

1998년 1월~현재 : 대한전자공학회 이사, 상임이사

2000년 4월~현재 : 한국이더넷 포럼 의장

1983년 8월~현재 : 한국전자통신연구원 네트워크연
구소 액세스기술연구부장, 책임연구원

<주관심 분야> 광 가입자망 기술, 고속 LAN 및 라
우터 기술, BISDN망, ATM 교환, 무선 및 이동