

임베디드 RG 를 위한 우선순위 버퍼관리 기법에 관한 연구

조은성, 유인태
경희대학교 전자계산공학과
경희대학교 전자계산공학과
e-mail : (escho,itryoo)@mic.khu.ac.kr

Priority-based Buffer Management in an Embedded Residential Gateway

Eun-Sung Cho, In-Tae Ryoo
Dept. of Computer Engineering, Kyung-Hee University
Dept. of Computer Engineering, Kyung-Hee University

요 약

Diffserv 에서 제시하고 있는 실시간 서비스에 대한 큐잉기법은 네트워크에서 사용자가 요구하는 다양한 서비스 특성을 배제하고 일반적인 우선순위를 할당하여 처리한다. 그러나, 이 기법은 실시간 서비스의 특징이 서로 다르고 요구되는 조건이 서로 다른 서비스에 대한 고려를 충분히 하고 있지 못하다. 따라서, 요구되는 조건이 서로 다른 다양한 서비스에 대한 서비스 품질(QoS; Quality of Service)를 보장하기 위하여, 기존의 버퍼관리 및 스케줄링 기법이 가지는 한계를 해결하기 위하여, 본 논문에서는 다양한 트래픽 특성에 따라 우선순위에 차등을 주어 효과적인 버퍼관리를 할 수 있는 방안을 제시하고 빠른 동작을 위하여 기존의 복잡한 연산을 간략화 함으로써 스케줄러의 성능을 높인다. 또한, 기존의 버퍼관리 및 스케줄러와 시뮬레이션을 통하여 성능을 평가하고, 제안하는 버퍼관리기법과 스케줄링을 적용하고 구현하기 위하여, 다양한 실시간 서비스와 비실시간 서비스를 수용하고 있는 임베디드 RG(Residential Gateway)에 제안하는 우선순위 버퍼관리와 스케줄러를 구현한다.

1. 서론

지난 몇 년간 초고속 망의 발전은 인터넷의 폭발적인 확산과 발전을 가져왔으며, 일반 사용자에게는 다양한 서비스를 사용할 수 있는 기회를 부여하였다. 이러한 추세는 현대에 들어 일반 사용자가 더욱 다양한 서비스를 요구하는 계기로까지 발전하였다. 서로 다른 서비스의 특성에 맞는 서비스 품질(QoS)를 보장하기 위하여 IETF[1]에서는 수년 전부터 Intserv[2]와 Diffserv[2]에 대한 권고안을 확정하였으며, 각 WG에서 각 서비스에 맞는 QoS를 보장하기 위한 연구가 진행되고 있다.

그러나, 차별화 된 서비스의 QoS를 보장하기 위한 Diffserv로는 서로 다른 특성을 갖는 실시간 서비스와 비실시간 서비스를 수용하기 위한 버퍼관리 및 스케줄링에 있어서 많은 문제점을 가지고 있다. 따라서, 서로 다른 특성을 갖는 실시간 서비스와 비실시간 서비스를 효과적으로 수용

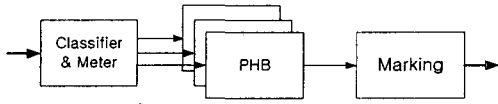
하기 위한 버퍼관리와 스케줄링이 필요하다.

본 논문에서는 실시간 서비스와 비실시간 서비스의 효과적인 버퍼관리 및 스케줄링을 위하여 정적 우선순위기반의 버퍼관리와 스케줄러의 방식을 제안·구현하며, 그 성능을 검증한다.

또한 다양한 실시간 서비스와 비실시간 서비스를 수용할 수 있고 Diffserv 의 기능을 구현할 수 있는 Linux 를 사용하여 임베디드 RG 에 적용한다.

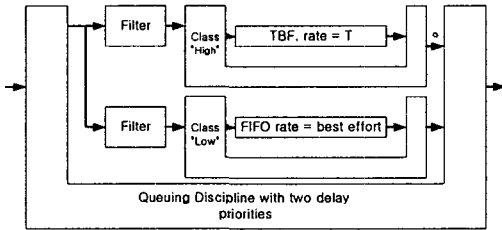
2. Diffserv 의 구조

IETF에서 규정된 Intserv[1][2]와 Diffserv[3][4]를 바탕으로 한 버퍼관리 및 스케줄러의 기능은 다음과 같이 구현되어 있다.



<그림 1> Diffserv forwarding path

Diffserv는 입력되는 데이터를 우선순위에 따라서 각각 클래스와 필터를 두어 처리하며, 서로 다른 서비스를 Expected Forwarding(EF), Assured Forwarding(AF), Best Effort(BF)로 나누어 PHB를 관리한다. 이러한 Diffserv에 바탕을 둔 Weight Fair Queuing(WFQ)기법의 동작 흐름도는 다음과 같다.



<그림 2> 우선순위, TBF, and FIFO 큐잉 방식

그러나 Diffserv는 단지 데이터의 우선순위와 대역폭을 임의로 할당하여 공평한 큐잉을 제공할 뿐, 각기 서로 다른 실시간 서비스를 수행하는데 효과적이지 못하고 음성이나 영상의 실시간 서비스를 수행하는데 문제점이 많다. 한편 과거의 Weighted Round Robin(WRR)과 Weighted Fair Queuing(WFQ), 그리고 1995년에 WRR과 WFQ를 보완한 Deficit Weighted Round Robin(DWRR)[5]은 버퍼와 대역폭, 그리고 스케줄링에 있어서 Deficit count를 적용하여 입력과 출력의 균형을 유지한다. 그러나 이러한 기존의 버퍼관리 및 스케줄링기법은 다음과 같은 문제점을 발생한다.

- 할당하는 대역폭과 우선순위에 대한 계산으로 인한 데이터 처리 효율저하
- 하드웨어로의 이식난해
- 높은 대역폭을 갖는 서비스에서의 정확성 저하

이러한 문제점을 해결하기 위하여 빠른 스케줄러와 효율적인 버퍼관리가 필요하다.

3. 우선순위 버퍼관리

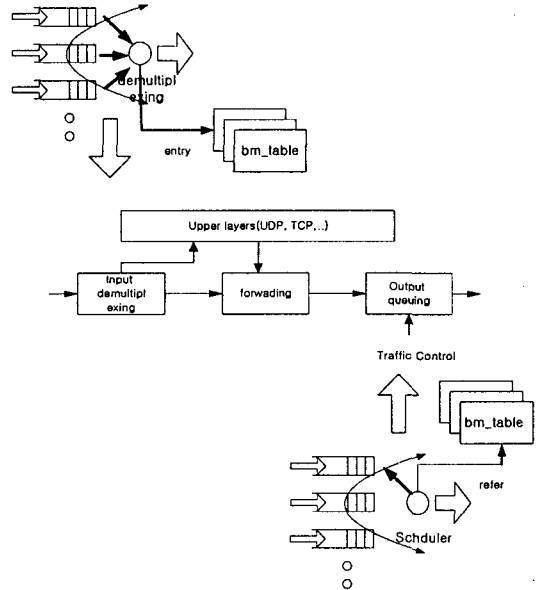
서비스를 제공하기 위하여 본 논문에서 고려하는서비스 품질(QoS)요소는 다음과 같다.

- 실시간 서비스 대역폭 보장
- 시간 지연의 최소화
- 실시간 서비스에서의 패킷간의 duration 보장
- 입력되는 데이터의 빠른 처리

실시간 서비스의 우선순위

본 논문에서 제안하는 우선순위버퍼관리는 서로 다른 특성을 갖는 실시간 서비스를 효과적으로 수용하고 낮은 우선순위를 갖는 최선을 다하여 대역폭을 제공하는 최선형(best effort) 서비스를 지원하기 위하여 입력되는 데이터의 특성에 따라 테이블을 관리하고, 각각의 서로 다른 특성을 갖는

서비스별로 버퍼 관리 테이블인 bm_table에 따라서 입력 데이터를 처리하고 스케줄링을 한다.



<그림 3> 정적 우선순위를 적용한 forwarding방법

제안 기법의 동작은 위의 그림에서와 같이 각 입력되는 패킷을 처리하는 입력 디멀티플렉서는 각각의 패킷의 서비스 타입에 따라서 버퍼관리 테이블에 그 패킷의 처리정보를 저장하는데, 이는 실질적으로 패킷 입력 카운터를 증가시키는 효과를 갖는다. 출력을 위한 스케줄러는 디멀티플렉서에서 참조된 카운터와 서비스의 임계값에 따라서 출력 대역폭을 결정하고 버퍼관리 테이블에 따라 스케줄링한다.

패킷을 처리하는데 걸리는 지연시간과 처리되는 버퍼의 크기를 결정하는 요인은 패킷의 크기보다는 입력되는 패킷의 갯수이다[6]. Voice of IP(VoIP)와 같은 주기적인 실시간 서비스는 일정한 주기를 갖는 서비스[7]이며, 영상 데이터와 같은 시간에 따라서 패킷간의 주기와 크기를 서로 같지 않은 서비스가 있다. 그러나 영상 서비스와 같은 비 주기성 실시간 서비스는 일정한 평균값을 갖는다. 이러한 서로 다른 특성의 실시간 서비스와 World Wide Web(WWW)과 같은 비 실시간 서비스의 버퍼관리와 스케줄링을 위하여 다음과 같은 버퍼관리 테이블을 구성한다.

```

bm_count {
    tm_count;
    threshold_value;
    buffer_size;
    average_value;
    min_value;
    max_value;
    time_duration;
    priority;
    * tm_table;
}
tm_table {
    priority[N]
    priority_base
  
```

```

    flag
}
N : constant value
    
```

디멀티플렉서에서의 동작

VoIP와 같은 주기적으로 일정한 실시간 서비스는 *bm_table*을 통하여 테이블의 주기적인 갱신없이 서비스 할 수 있다. 그러나 영상 서비스와 같은 비주기성 실시간 서비스는 실시간 서비스에 대하여 일정한 시간을 가지고 입력되는 패킷에 따른 *bm_table*을 수행하여야 한다. 입력되는 패킷의 개수에 따른 단순한 방법의 연산을 통하여, 기존의 복잡한 계산에 따른 부하를 갖지 않고 비주기성 실시간 서비스를 보장한다.

디멀티플렉서에 할당되는 버퍼의 크기는 패킷간의 주기와 크기에 따라 계산된다. 그 계산된 버퍼의 크기는 *bm_table*의 *buff_size*에 기록되고 실시간 서비스에 대한 추가 요구가 발생한 경우에 대역폭에 대한 증가와 감소의 요소로 작용한다. 할당 받은 *bm_table*과 버퍼에 대한 서비스가 종료될 때까지 계속되며, 버퍼에 대한 요구사항이 발생할 경우에 재시동이 가능하다.

본 논문에서 제안하는 정적 우선순위를 이용한 버퍼관리 및 스케줄러는 서비스에 따라서, *bm_table*에 기록된 서비스의 특성에 따른 버퍼를 할당하고 우선순위를 지정하며, 서비스되는 패킷의 주기와 크기에 대한 최소 측정치와 최대 측정치를 계산하여 평균치에 해당하는 평균값을 주기적으로 계산하여 우선순위에 반영한다.

$C_i(t) = \{ C_i(1), C_i(2), C_i(3), \dots, C_i(N) \}$,
 $Cr(t) = \{ Cr(1), Cr(2), Cr(3), \dots, Cr(N) \}$,
 C_i : t시간의 입력 패킷의 개수
 Cr : 일정 주기에 입력 패킷의 개수
 N : constant value

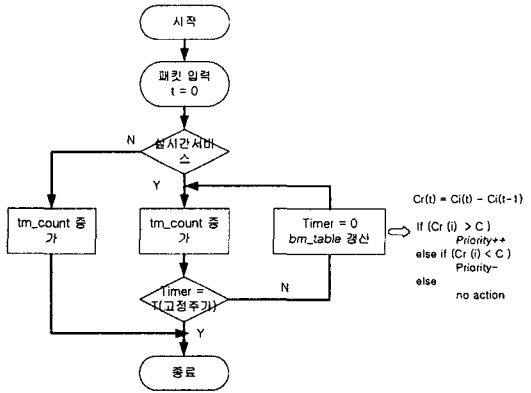
$Cr(t) = C_i(t) \square C_i(t-1)$

```

If (Cr (i) > C )
    Priority++ # priority of bm_table;
else if (Cr (i) < C )
    PriorityD
else
    no action
    C : 평균 입력 패킷수
    
```

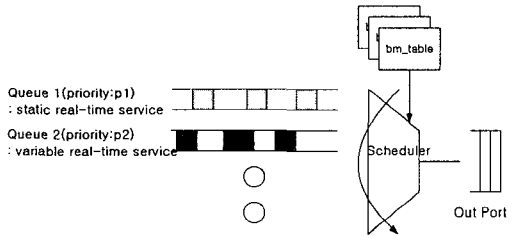
상기 식을 이용하여 입력되는 패킷의 양에 따라서 버퍼에 유입되는 패킷의 증가값을 알 수 있다. 또한 점차 증가하는 실시간 서비스에 대하여 스케줄러에게 버퍼에 대한 점유율과 스케줄링에 대한 정보를 제공한다.

입력되는 패킷의 우선순위는 주어진 임계값에 따라서 할당할 수 있으나[8], Packet의 개수가 주기적으로 일정한 실시간 서비스와 비주기적으로 변하는 실시간 서비스간에 우선순위의 구별을 둔다.



<그림 4> 디멀티플렉서의 동작

스케줄러는 디멀티플렉서에서 기록된 *bm_table*에 기록된 정적 우선순위에 따라서 주기적으로 패킷의 크기와 주기가 일정한 실시간 서비스, 패킷의 크기와 주기가 변하는 실시간 서비스, 그리고 비실시간 서비스를 스케줄링한다.



<그림 5> *bm_table*을 이용한 scheduler

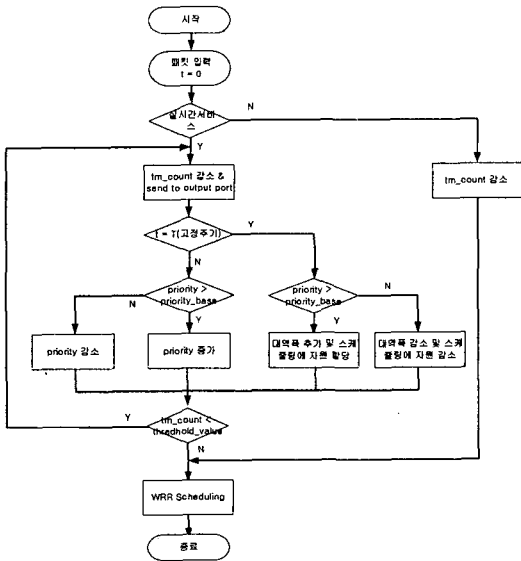
위의 그림에서 스케줄러는 디멀티플렉서에서 기록된 *bm_table*에 따라 *tm_count*를 참조하여 초기에 설정된 *tm_count*값이 일정한 임계값을 초과한 경우에, 임계값에 맞추기 위하여 초과된 값만큼의 패킷을 우선순위에 따라서 처리하며, 비주기성 실시간 서비스는 *bm_table*의 *max_value*을 초과한 경우 우선순위에 따라서 처리한다. 일정하지 않은 실시간 서비스에 대하여 *bm_table*이 지시하는 *tm_table*의 우선순위를 참조하여 대역폭과 스케줄링에 참조한다.

```

if ( priority > priority_base )
    ++(bandwidth and give a scheduling time)
else ( priority < priority_base )
    --(bandwidth and scheduling time)
    
```

Queue에 적용되는 p1,p2는 각각 *bm_table*의 priority를 나타낸다.

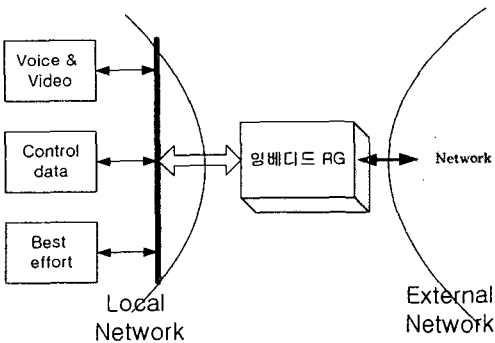
초기에 할당하는 버퍼는 VoIP와 같은 주기적인 실시간 서비스는 WRR에서와 같이 일정한 버퍼를 할당한다. 그러나 영상 신호와 같은 크기가 주기적으로 일정하지 않는 데이터는 위의 *bm_count*의 *average_value*에 따라서 그 대역폭을 *min_value*와 *max_value*에 따라서 주기적으로 갱신한다. 스케줄러는 일정주기의 실시간 서비스는 일정한 주기로 수행하며 주기적이지 않은 실시간 서비스는 그 *average_value*에 따라서 주기적으로 운용한다.



<그림 6> 스케줄러의 동작 원리

임베디드 RG에서의 우선순위 버퍼관리

다양한 실시간 서비스와 비실시간 서비스를 제공하는 임베디드 RG에 그 기능을 구현하여 적용한다. 제안하는 선형 우선순위 버퍼관리 및 스케줄링을 구현하기 위하여, Source를 쉽게 구할 수 있는 linux에서 그 기능을 구현하였다.



<그림 7> 우선순위를 갖는 임베디드 RG

위의 그림에서와 같이 WRR 과 WFQ 에 관련된 버퍼관리 및 스케줄링을 임베디드 RG 에서 구현하고 시험하였으며, 기존의 WRR 과 WFQ, 그리고 본 논문에서 제시하는 정적 우선순위 기법을 수용한 임베디드 RG 에 대한 기능을 소프트웨어로 구현하여 성능을 검증한다.

4. 결론

우선순위 버퍼관리를 이용하여 주기적인 실시간 서비스 뿐만 아니라 주기적으로 크기가 변하는 실시간 서비스를 효과적으로 수용하는 방안을 제시하였다. 기존의 실시간 서비스는 버퍼관리 및 스케줄러를 일관적으로 처리하였으나, 각각의 실시간 서비스는 특성이 서로 다르다. 이러한 서로 다른 특성을 수용할 수 있도록 제안한 정적 우선순위 버퍼관리 기법은 Buffer Management Table 을 이용하여 효과적인 버

퍼관리 및 스케줄링을 수행하도록 하였다. 그로 인하여 기존의 복잡한 연산으로 인한 스케줄러의 효율을 향상시켰으며 사용자의 정보를 저장하는 버퍼의 성능을 향상하였다. 각기 다른 특성을 가지고 있는 실시간 서비스에 대하여, 정적 우선순위 버퍼관리와 같은 망의 효과적인 버퍼 자원의 분배 및 스케줄링이 가능한 실시간 서비스의 버퍼관리 및 스케줄링에 대한 지속적인 연구가 필요하다.

참고문헌

- [1] "Integrated Service in the Internet Architecture: an Overview", RFC1633 : <http://www.ietf.org>
- [2] "A Framework for Integrated Services Operation over Diffserv Networks", RFC2998 : <http://www.ietf.org>
- [3] "An Architecture for Differentiated Services", RFC2475 : <http://www.ietf.org>
- [4] "Definition of Differentiated Service Per Domain Behaviors and Rules for their Specification", RFC3086 : <http://www.ietf.org>
- [5] "Efficient Fair Queuing Using Deficit Round Robin", ACM SIGCOMM' 95, vol.35, no.4(April1987):435-438
- [6] "The Cost of QoS Support in Edge Devices An Experimental Study", R.Guerin, L.Li, S.Nadas, P.Pan, V.Peris; In Watson Research Center
- [7] "실시간 처리 리눅스 기반 VoIP 시스템 설계 및 구현", 2001 정보처리학회 봄 학술발표논문집(A)
- [8] "Pricing Differentiated Services Networks: Bursty Traffic", Peter Marbach: IEEE INFOCOM 2001
- [9] "Linux Network Traffic Control - Implementation Overview", Werner Almesberger, EPFL ICA; 1999년 5월 23일
- [10] "Differentiated Services on Linux", Werner Almesberger, EPFL ICA; Jamal Hadi Salim, CTL Nortel Networks; Alexey Kuznetsov, INR Moscow; 1999년 7월 25일
- [11] "Embedded Linux Hardware, Software, and Interfacing", Craig Hollabaugh: Addison Wesley