

EPON 망에서 MPCP 프로토콜 기반의 RC-DBA 패킷 스케줄링 알고리즘의 FPGA 구현 및 임베디드 리눅스 기반의 검증 시스템 개발

강현진 · 장종욱

동의대학교

FPGA Implementation for packet scheduler through a RC-DBA algorithm and Development for verification system on Embedded Linux

Hyun-jin Kang · Jong-wook Jang

Donguei University

E-mail : khj626@deu.ac.kr · jwjang@deu.ac.kr

본 연구는 BB21과 지방대학 인력양성사업 과제에 의해 지원되었음.

요 약

EPON의 상향 전송 방식에서는 다수의 ONU가 OLT로부터 공유된 채널에 대한 권한을 할당받아 데이터를 전송하게 되므로 EPON에서는 각각의 ONU들에게 공유된 대역폭을 공평하고 효율적으로 할당하기 위한 DBA 알고리즘이 필수적이다. 우리는 본 논문에 앞서 기존의 DBA 알고리즘들의 문제점을 보완하여 Request-Counter Dynamic Bandwidth Assignment 알고리즘을 새롭게 제안하여 성능평가 및 비교 분석을 하였다. 본 논문에서는 제안된 RC-DBA 알고리즘을 적용하여 OLT의 MAC 스케줄러를 설계하고 Corebell 사의 LDS2000 FPGA ver.1.0 보드에 구현하였다. 또한 이를 검증하기 위해서 임베디드 리눅스 기반의 검증 시스템을 개발하였다.

키워드

Ethernet-PON, Request-Counter Dynamic Bandwidth Assignment, Embedded Linux, FPGA, Device Driver

1. 서론

EPON은 하나의 OLT에 다수의 ONU가 연결되는 점대다점 구조를 가지고 있으므로 그림 1과 같이 ONU에서 OLT로 데이터를 전송하는 상향 전송에서는 광 매체를 공유하게 된다. 이때 모든 ONU들은 공유된 매체를 효율적이고 공평하게 사용할 수 있어야 한다.

IEEE802.3ah EFM에서는 동적 대역폭 할당의 핵심이 되는 패킷 스케줄링 알고리즘에 관한 부분은 표준화 대상에서 제외하고 있다. [1]의 논문은 기존의 여러 가지 DBA 알고리즘의 단점을 보완하여 RC-DBA라는 새로운 스케줄링 알고리즘을 제안했으며, 본 논문의 FPGA 설계를 위한 이론적 바탕이 된다.

본 논문에서는 하드웨어 구현 언어인 Verilog HDL로 작성된 RC-DBA 기반의 OLT MAC 스케줄러[2]를 ALTERA 사의 QuartusII tool을 이용해 Gate-level로 합성 및 Placement & Route

하고, 동일한 업체의 Cyclone EP1C12F324C8 FPGA 칩에 구현하였다.

임베디드 리눅스를 이용한 검증 시스템은 Intel PXA255 프로세서를 탑재한 Corebell 사의 LDS2000 FPGA ver.1.0 시스템으로 리눅스 커널 2.4.18 버전을 운영체제로 사용한다. 리눅스 PC 시스템과의 Cross Compile 환경에서 디바이스 드라이버 및 검증 어플리케이션 프로그램을 구현하였다.

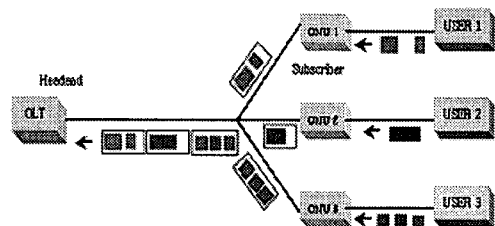


그림 1. EPON의 상향 전송 방식

논문의 구성은 2장에서 RC-DBA 알고리즘에 대해서 알아보고 3장에서는 RC-DBA 알고리즘을 적용한 OLT 스케줄러와 임베디드 리눅스와의 연결을 위한 CPU interface logic의 설계 및 구현에 대해서 다룬다. 4장에서는 디바이스 드라이버와 응용 프로그램을 구현하고 검증 프로그램을 수행하였다. 마지막으로 5장에서 본 논문의 결론을 맺는다.

2. RC-DBA 알고리즘

기존에 나와 있는 여러 가지 DBA 알고리즘들이 있지만 본 논문에서는 우리가 새롭게 제안한 RC-DBA 알고리즘을 사용하여 OLT의 스케줄러를 구현하였다. 일반적으로, DBA 알고리즘은 다수의 우선순위 큐 지연, ONU들 사이의 공평한 대역할당, 프레임 지연의 최소화 등의 기능을 가져야 한다. 기존의 DBA 알고리즘들은[3, 4, 5, 6, 7] 다수의 우선순위 큐를 지원하여 프레임 지연 시간을 최소화하는 기능은 뛰어나지만, 각각의 ONU들에게 공평하게 대역을 할당하는 기능은 제공되지 않는다. 예를 들면, ONU에서 각각의 트래픽 클래스별로 현재 요청한 대역폭과 전체 남아 있는 대역폭의 비율을 계산하여 상향 대역폭을 할당하는 기존의 ETRI DBA 알고리즘은[3] 동일한 우선순위 큐에 대해서 전체 할당할 수 있는 상향대역폭에 현재 요청한 양을 비례적으로 할당한다. 이러한 경우, 특정 ONU가 계속해서 다른 ONU에 비해 상당히 많은 양의 대역폭을 요구하게 되면 다른 ONU가 대역폭을 보장받지 못하는 경우가 발생할 수 있다. RC-DBA 알고리즘은 이러한 문제를 해결하기 위하여 동일한 우선순위일 경우, 각 ONU별로 가중치를 부여하여 가중치가 높은 순서대로 ONU에게 상향 대역폭을 할당하는 메커니즘을 사용한다[8].

RC-DBA 알고리즘에서는 다음과 같은 기준에 의해 가중치가 부여된다. 먼저, OLT는 상향 대역폭을 요청한 ONU와 그렇지 않은 ONU를 구분하여 대역폭을 요청한 ONU에게만 전체 ONU의 수만큼 가중치를 증가시킨다. 그리고 상향 대역폭을 많이 요청한 ONU 순으로 가중치를 부여하는데, 예를 들어 전체 ONU 수가 16개이고 현재 요청한 ONU가 모두 10개라면 상향 대역폭을 가장 많이 요청한 ONU에게는 해당 가중치 값에 10을 증가시키고 그 다음 순으로 9, 8, 7, ..., 2, 1씩 증가시킨다. 첫 번째 단계에서 가중치 값을 증가시킬 때 단순히 1씩 증가시키는 것이 아니라 전체 ONU 수만큼 증가시키는 이유는, 현재 상향 대역폭에 대한 요청을 했으나 대역폭을 할당받지 못한 ONU가 다음 report 메시지를 전송한 후에 상향 대역폭을 할당받을 우선순위가 다른 ONU보다 높아야 하는데 단순히 1만 증가시키면 그렇게 되지 못하는 경우가 발생하여 모든 ONU들이 공평한 대역폭 할당을

받지 못하기 때문이다[9].

3. RC-DBA 알고리즘을 적용한 OLT 스케줄러와 PXA255 Interface logic의 설계

설계된 OLT 스케줄러의 동작은 크게 네 부분으로 나뉜다. 각각의 ONU들의 우선순위 큐의 상태, 즉 대역폭 요청량 정보를 담은 report 메시지를 받는 부분, 수신된 report 메시지의 요청량 정보를 RC-DBA 알고리즘에 적용하여 가중치를 계산하는 부분, 계산된 가중치 값에 비례해 대역폭을 할당하는 부분, 마지막으로 대역폭 할당 정보를 grant 메시지로 만들어 각 ONU에게 전송하는 부분이다.

그림 2는 2장에서 설명한 RC-DBA 알고리즘의 동작, 즉 대역폭 요청량에 따른 가중치 값을 구하는 부분의 ASM 차트이다. Verilog HDL로 구현된 스케줄러는 [2]의 논문에서 시뮬레이션 프로그램인 modelSim으로 1차 기능검증이 되었다.

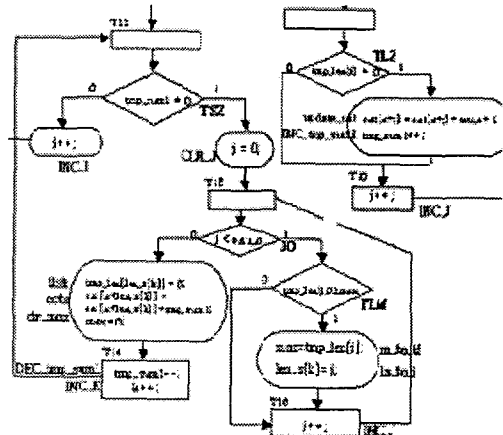


그림 2. RC-DBA 알고리즘의 ASM 차트

OLT 스케줄러 모듈은 임베디드 리눅스 기반의 디바이스 드라이버와 응용 프로그램에 의해서 제어되는데, 이를 위하여 CPU interface logic 부분이 추가되었다.

Interface logic은 CPU로부터 받은 데이터를 저장하는 레지스터들의 집합이다. 스케줄러에서는 48bit 크기의 report와 grant 메시지를 사용하는데, LDS2000 임베디드 시스템과 FPGA 보드 사이의 입출력 버스는 32bit이다. 그러므로 48bit 크기의 report 메시지 한 개를 입력받기 위해서는 디바이스 드라이버에서 16bit씩 데이터를 나누어 3번에 걸쳐 interface logic으로 보내주고, interface logic에서는 이것을 48bit report 데이터로 다시 조합하여 스케줄러 모듈에서 사용하게 된다. grant 메시지 출력에서도 동일하게 16bit씩 데이터를 분리하여 출력하고 수신측인 CPU에서 재조합하게 된다. 그림 3은 PXA255 CPU와

FPGA 사이에서 데이터 입·출력을 위해 필요한 인터페이스 제어신호이다.

FPGA 칩으로 합성된 CPU interface OLT 스케줄러는 FPGA 전용 합성 툴인 Altera사의 QuartusII를 사용하여 gate-level로 합성하였다. 합성된 스케줄러는 Cyclone EP1C12F324C8 FPGA에서 24%(2,938/12,060)의 logic cell로 합성 및 배선·배치되었으며, Registers는 10%(1,349/12,795)를 사용하였고 Actual fmax는 62.03MHz였다.

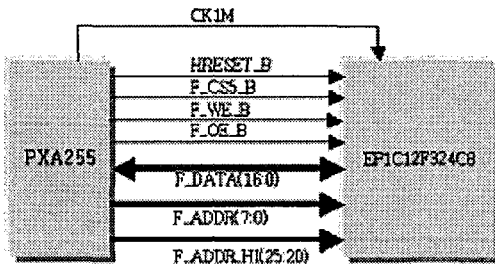


그림 3. CPU와 FPGA의 제어신호 인터페이스

4. 임베디드 리눅스 기반의 검증시스템

3장에서 잠시 언급했듯이 OLT 스케줄러는 임베디드 리눅스 기반의 시스템을 통해서 제어되는데, 이것은 디바이스 드라이버라는 장치 제어 프로그램으로 가능하다. 특정 장치를 사용하여 원하는 작업을 처리하려는 사용자는 먼저 장치와 관련된 디바이스 드라이버를 제작하고 이후에 응용 프로그램을 작성하여 원하는 작업을 수행하게 된다. 즉, 디바이스 드라이버는 장치와 데이터 입·출력 과정을 수행하고, 응용 프로그램에는 사용자가 궁극적으로 원하는 처리 작업을 작성한다[10].

디바이스 드라이버는 장치 제어를 목적으로 커널 차원에서 제공되는 기능으로서 제어할 장치의 종류에 따라 블록 디바이스 드라이버, 네트워크 디바이스 드라이버, 문자 디바이스 드라이버로 구분되는데[10], 본 논문에서는 문자 디바이스 드라이버를 통하여 FPGA 보드를 제어하게 된다.

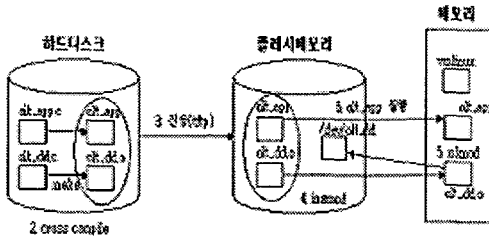


그림 4. device driver와 응용프로그램의 구동과정[11]

그림 4는 디바이스 드라이버와 응용프로그램의 컴파일부터 실행까지의 구동 과정을 나타낸다. 임베디드 시스템은 제한적인 특성으로 인해 프로그램 개발환경을 자체적으로 지원해 줄 수 없기 때문에 일반 리눅스 PC에서 프로그램 작성 및 컴파일을 한다. 그림의 하드디스크 내부는 일반 리눅스 PC에서 디바이스 드라이버 olt_dd.c와 응용프로그램 olt_app.c를 작성하고 크로스 컴파일러를 통해 임베디드 리눅스용의 오브젝트 파일과 실행파일을 만드는 과정을 나타낸다. 생성된 모듈과 실행파일은 실제 사용하게 될 임베디드 시스템으로 전송하게 된다. FPGA칩의 스케줄러 장치를 사용하기 위해서는 먼저 모듈을 임베디드 리눅스에 insmod하여 임베디드 시스템의 커널 메모리에 적재하고 mknod를 통하여 OLT 스케줄러(장치)를 등록한다. 그리고 사용자는 응용프로그램을 통하여 OLT 스케줄러 장치를 테스트할 수 있다.

그림 5, 6은 임베디드 리눅스에서 olt_dd.o 모듈의 적재 및 olt_app의 실행 화면이다.

```

root@be-lab:/home/DS/EPON/ols/ols/ols# ./olt_app
bash-2.05b# insmod olt_dd.o
Using olt_dd.o
Loading OLT Device Driver...
OLT Device Driver registration OK with major number = 253
bash-2.05b# mknod /dev/olt_dd c 253 0
bash-2.05b# ls -l /dev
crw-rw-rw- 3 0 0 4096 .
crw-rw-rw- 13 0 0 4096 .
crw-rw-rw- 1 0 0 254 0 console
crw-rw-rw- 1 0 0 5 1 console
crw-rw-rw- 1 0 0 0 initctl
crw-rw-rw- 1 0 0 1 2 kmem
crw-rw-rw- 1 0 0 1 1 kmem
crw-rw-rw- 1 0 0 1 3 null
crw-rw-rw- 1 0 0 253 0 olt_dd
    
```

그림 5. 디바이스 드라이버 적재

```

root@be-lab:/home/DS/EPON/ols/ols/ols# ./olt_app
bash-2.05b# ./olt_app
OLT open with major/minor (253 / 0)
OLT FPGA detected.

***** main menu *****
* w. write 5 report packets *
* r. read 5 grant packets *
* *
* 1. FPGA reset *
* 2. OLT's Scheduler run *
* 3. FPGA quit *
*****

select the command number : 1

reset board
    
```

그림 6. 응용프로그램 실행

5. 결론

IEEE802.3ah EFM에서 대역할당정책이 표준화 대상에서 제외된 이유는 EPON 서비스를 제공하는 업체에 유연성을 제공하기 위함이다. 이것은 다수의 장비제조 업체 간의 장비운용 호환성의 문제가 대두될 수 있지만, 업체별 제품성능의 차별화를 내세울 수 있다.

RC-DBA 알고리즘은 모든 ONU에게 공평한 대역할당을 하지 못하는 기존의 DBA 알고리즘들의 문제점을 해결하면서 QoS를 지원하기 위하여 MPCP를 이용하여 설계되었다.

본 논문에서는 RC-DBA 알고리즘을 적용한 OLT 내부의 대역할당 스케줄러를 FPGA 칩에 구현하고 이를 위한 디바이스 드라이버 모듈 및 검증 프로그램을 임베디드 리눅스 기반에서 개발하였다.

본 연구는 DBA의 핵심이 되는 패킷 스케줄링 알고리즘을 하드웨어로 구현함으로써, 차후 OLT 내부의 MAC 칩으로 확장 구현할 수 있다.

본 연구과제의 최종 목표는 OLT에게 report 메시지를 전송하고 OLT의 grant 메시지를 수신할 수 있는 ONU도 FPGA 칩에 구현해서 실제 OLT와 ONU 사이에 통신을 통하여 RC-DBA 알고리즘의 동작을 확인하는 것이다. 이를 위해서 우리는 OLT 스케줄러와 관련된 동작을 수행하는 ONU 시스템도 설계 중에 있다.

참고문헌

[1] 장성호, "EPON망에서 ONU 공정성을 고려한 RC-DBA 알고리즘의 설계와 성능평가", 동의대학교 컴퓨터공학과 박사학위 논문, 2004

[2] 강현진, 장종욱, "EPON망에서 RC-DBA 알고리즘을 적용한 OLT MAC 프로토콜의 구현", 한국해양정보통신학회 2006 동계종합학술대회, pp.66-70, 2006

[3] S. Choi and J. Huh, "Dynamic Bandwidth Allocation Algorithm for Multimedia Services over Ethernet PONs," ETRI Journal, Vol. 24, No. 6, pp. 465-468, Dec. 2002.

[4] H. Shimonishi, I. Maki, T. Murase, and M. Murata, "Dynamic Fair Bandwidth Allocation for Diffserv Classes," Proceeding of IEEE ICC, Vol. 4, pp. 2348-2352 Apr.-May 2002.

[5] Chadi M. Assi, Yinghua Ye, Sudhir Dixit and Mohamed A. Ali, "Dynamic Bandwidth Allocation for Quality-of-Service Over Ethernet PONs," IEEE Journal on Selected Areas in

Communications, Vol. 21, No. 9, Nov. 2003.

[6] J. Moon, J. Park, and M. Lee, "Hybrid Bandwidth Allocation Algorithm To Support Multiple Services in Ethernet PON," Proceeding of ICACT 2003, pp. 692-696, Jan. 2003.

[7] Fu-Tai An, Yu-Li Hsueh, Kyeong Soo Kim, Ian M. White, and Leonid G. Kazovsky, "A New Dynamic Bandwidth Allocation Protocol with Quality of Service in Ethernet-based Passive Optical Networks," Proceeding of IASTED WOC 2003, pp. 383-395, Jul. 2003.

[8] Seng-Ho Jang and Jong-Wook Jang, "New DBA Algorithm Supporting QoS for EPON", The CS&CE International Multiconference on CIC 2004, Las Vegas, USA, Jul. 2004.

[9] Seong-Ho Jang and Jong-Wook Jang, "Performance Evaluation of a New DBA Algorithm Supporting Fairness and Priority for Ethernet-PON", the IASTED International Conference on OCSN 2004, Banff, Canada, Jul. 2004.

[10] (주) 코어벨, "리눅스 디바이스 드라이버의 활용", pp. 105-140, 2003

[11] 박성일, "몽고메리 곱셈기의 FPGA 구현 및 임베디드 리눅스 기반 검증 시스템 개발", 동의대학교 컴퓨터공학과 석사학위 논문, pp. 38, 2004