

RC-DBA 알고리즘을 적용한 OLT와 ONU의 Verilog HDL 구현

강현진^o 장종욱

동원대학교 컴퓨터·소프트웨어 공학과
{khj626^o, jwjang}@deu.ac.kr

Implementation of a OLT and ONU through a RC-DBA algorithm by verilog HDL

Hyunjin Kang^o Jongwook Jang
Donguei University Computer-Software Engineering

요 약

EPON은 점대다점 구조를 가지기 때문에 ONU에서 OLT로 데이터를 전송하는 상향 트래픽 전송에서 전송매체를 공유함에 따라 충돌이 발생하게 된다. 그러므로 모든 사용자가 공정한 매체 접근 권한을 가지고 경쟁 없이 전송을 하기 위한 매체접근 제어 기술이 필수적이다. RC-DBA 알고리즘은 버스트 트래픽 지원과 ONU들 사이의 공정성 지원 등의 장점을 가진 매체접근 제어 알고리즘이다. 본 논문에서는 RC-DBA 알고리즘을 적용한 OLT와 간단한 ONU를 하드웨어 기술 언어인 Verilog HDL을 사용하여 작성하고 시뮬레이션을 해봄으로써 그 기능을 검증하였다.

1. 서 론

EPON에서는 ONU에서 OLT로 데이터를 전송하는 상향 트래픽 전송에서 전송매체를 공유함에 따라 충돌이 발생하게 된다. RC-DBA(Request-Counter Dynamic Bandwidth Assignment) 알고리즘은 IEEE 802.3ah 그룹에서 표준으로 제정한 MPCP(Multi-Point Control Protocol)를 응용하여 공유된 채널의 용량이나 자원을 보다 효율적으로 사용할 수 있도록 제안된 MAC 프로토콜 알고리즘으로서 버스트 트래픽 지원, ONU들 사이의 공정성 지원 등 여러 가지 장점들을 가지고 있다[1]. 본 논문에서는 이러한 RC-DBA 알고리즘을 적용한 OLT와 자신의 큐 정보를 알리기 위한 Report 메시지를 출력하는 간단한 ONU를 하드웨어 기술언어인 Verilog HDL을 사용하여 코드를 작성하고 시뮬레이션을 통해 기능검증을 수행하였다. OLT와 ONU는 각각 시뮬레이션을 위한 설계사양을 결정하고 ASM 차트와 상태표를 작성하는 과정을 통해 코드가 작성되었다[2, 3, 4]. 본 논문의 구성은 2장에서는 시뮬레이션을 위한 설계사양에 대해서 설명하고 3장과 4장에서는 각각 OLT와 ONU의 구현과 시뮬레이션 결과에 대해서 설명한다. 마지막으로 5장에서 결론을 내도록 한다. 디버깅 및 시뮬레이션 도구로는 Model Tech사의 Modelsim EE/Plus 5.2c 프로그램을 사용하였다.

2. 설계 사양

다음은 OLT와 ONU의 기본적인 설계 사양이다.

- ① ONU 내부의 우선순위 큐의 개수는 3개이다. (high, middle, low)
- ② Report와 Grant 메시지의 크기는 총 48비트이다.

- Report는 OLT 시스템의 외부 입력으로 우선순위 큐의 요청량 정보를 포함한다.
- Grant는 OLT 시스템의 외부 출력으로 우선순위 큐에 대한 대역할당 정보를 포함한다.

그림 1은 MAC 기능을 수행하기 위한 Report와 Grant 메시지의 형식이고 그림 2은 ONU로 입력되는 Ethernet 프레임과 IP 헤더의 TOS byte의 형식을 나타낸 것이다. 실제 EPON에서 사용하는 메시지 포맷 중에서 시뮬레이션에 필요한 최소한의 정보만을 포함하고 나머지는 생략하였다.

| | | | | | | |
|-----------|-----------|--------------|---------------|-------------|---------------|------------|
| DA (4) | SA (4) | Q_NUM (8) | Bitmap (8) | High (8) | Middle (8) | Low (8) |
|-----------|-----------|--------------|---------------|-------------|---------------|------------|

< REPORT 메시지 >

| | | | | | |
|-----------|-----------|-----------------------|------------------|--------------------|-----------------|
| DA (4) | SA (4) | Opcode(00-02) (16) | High grant(8) | Middle grant(8) | Low grant(8) |
|-----------|-----------|-----------------------|------------------|--------------------|-----------------|

< GRANT 메시지 >

그림 1. Report, Grant 메시지의 형식

| | | | | | | |
|-----------|-----------|-----------------------|------------|-----------|-----------|-------------|
| DA (4) | SA (4) | Etype(0x0800) (16) | TOS (8) | DA (4) | SA (4) | Data (8) |
|-----------|-----------|-----------------------|------------|-----------|-----------|-------------|

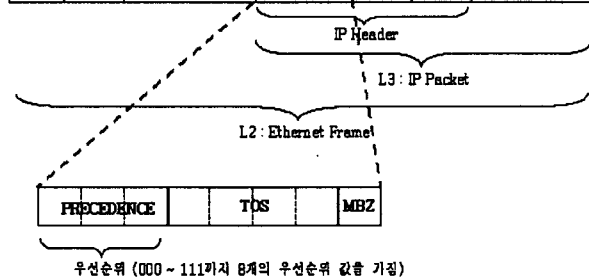


그림 2. Ethernet 프레임과 TOS byte의 형식[5]

ONU에 입력된 프레임들은 분류자(Classifier)에서 우선 순위 별로 분류된 후, 해당 우선순위 큐에 저장된다. 분류자는 IEEE 802.1p표준의 우선순위 필드를 확인하고 IEEE 802.1p의 우선순위 정보는 상위 레이어 프로토콜인 IP 패킷의 TOS(Type of Service) 필드 정보를 이용하여 얻을 수 있다[5, 6]. 표 1은 802.1p의 트래픽 형식에 따른 802.1p의 value를 나타내고 각각의 우선순위는 임의로 결정하였다.

표 1. IEEE 802.1p[7]

| Value | Traffic 형식 | Priority |
|-------|----------------------|----------|
| 0 | background | low |
| 1 | spare | low |
| 2 | Best Effort | low |
| 3 | Business critical | medium |
| 4 | streaming multimedia | medium |
| 5 | 지연속도가 100ms이하인 음성 | medium |
| 6 | 지연속도가 10ms이하인 음성 | high |
| 7 | 네트워크 제어 트래픽 | high |

3. OLT의 구현

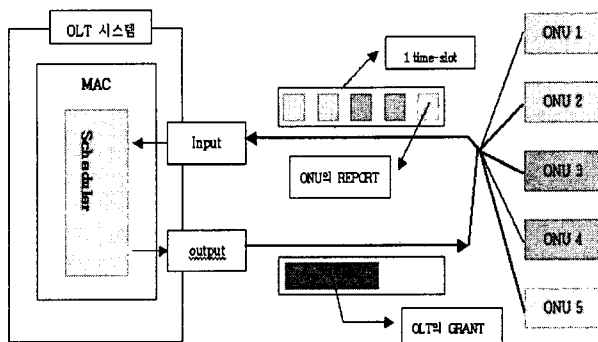


그림 3. 전체적인 OLT의 동작 블록도[6]

그림 3은 OLT 시뮬레이션의 전체적인 동작을 블록도로 나타낸 것이다. OLT는 다섯 개의 ONU에서 보낸 Report 메시지의 우선순위 큐들의 요청량 정보를 사용하여 Scheduler에서 우선순위 큐들의 대역폭을 계산하고, 계산된 대역폭 할당 정보를 Grant 메시지에 실어서 ONU들에게 브로드 캐스팅한다. 이러한 OLT시스템의 동작은 크게 세 부분으로 나눌 수 있는데 첫째, Report 메시지 입력, 둘째, 우선순위 큐의 요청량에 따른 가중치 계산, 셋째, 대역폭 할당 및 Grant 메시지 출력이다. 모든 프로그램은 start 신호에 1이 입력되면 시뮬레이션의

동작이 시작되고, 모든 과정이 끝나면 제어회로는 done 신호에 1을 인가하여 시뮬레이션의 동작이 완료되었음을 알린다. ASM차트 및 상태표 작성, Verilog HDL 구현 등의 모든 과정은 각 부분별로 나눠져 구현되었다.

각 부분에 대한 verilog HDL 코드는 modelsim에서 디버깅하고 시뮬레이션 하였다. 그림 4는 Report 메시지 입력 부분을 코딩한 후 Test bench를 생성하여 시뮬레이션한 결과파형이다. ①은 Report 메시지가 OLT로 차례로 입력되는 모습을 보여주고 있다. r_to_p 신호는 Report 메시지가 OLT의 레지스터로 입력되기 위한 제어 신호이다. ②에서 ④까지는 Report 메시지의 요청량 정보를 추출해 내는 부분이다. 동일한 우선순위내의 모든 ONU가 요청한 상향 대역폭 양을 Report 메시지에서 추출하여 새로운 레지스터인 length에 저장하는 것이다. length에 저장하기 위한 제어신호는 l_to_RH, l_to_RM, l_to_RL이다. 그림 5는 RC-DBA 알고리즘의 가중치 계산 부분의 결과 파형이다. 가중치 계산은 당연히 우선순위 큐별로 이루어진다. ①은 high priority queue들 중 상향 대역폭 요청량이 있는 큐들의 가중치를 전체 ONU 수만큼 증가시킨다. update_counter신호는 가중치를 증가시키기 위한 제어신호이다. ②는 high priority queue의 가중치 계산의 두 번째 단계인 요청량이 많은 순서대로 가중치를 계산하는 부분이다. 가장 요청량이 많은 ONU의 high priority queue는 대역폭을 요청한 ONU의 총 개수를 현재의 가중치 값에 더하고 이후에는 요청량이 많은 순서대로 1씩 감소시킨 후에 가중치 값에 더한다[8]. ccts는 두 번째 단계에서의 가중치 증가 제어 신호이다. ③과 ④는 middle과 low priority queue들의 가중치 계산 부분인데 시뮬레이션에서는 middle과 low 큐들의 대역폭 요청량이 없으므로 가중치 계산 알고리즘을 적용시키지 않는다. 즉, update_counter와 ccts 신호에 반응이 없다. 마지막으로 대역폭 할당은 per_to_len과 per_to_tbw 신호에 의해 알 수 있다. 시뮬레이션을 두 번 이상 수행하게 된다면 현재 대역폭을 할당받지 못한 큐들은 두 번째 동작에서 대역폭을 할당받게 된다. 대역폭 할당이 완료되면 make_grant신호가 1이 되면서 Grant 메시지를 생성하여 출력하게 된다. Grant 메시지는 임의의 레지스터에 필요한 정보들이 저장됨과 동시에 출력포트로 내보내지게 된다.

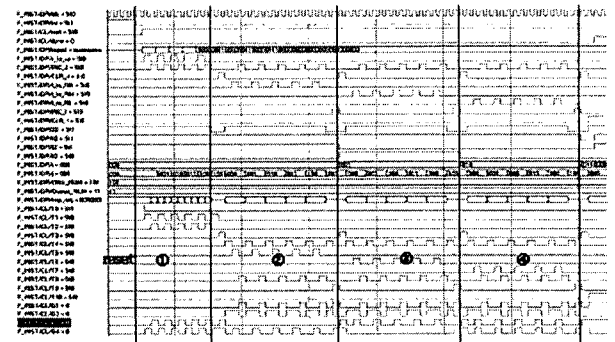


그림 4. Report 메시지 입력의 시뮬레이션 결과 파형

