

ATM Adaptation Layer Type 2 스위칭을 위한 소프트웨어 스위치 서버 구현

○ 신 동훈, 김 태석, 권 태근
 충남대학교 컴퓨터공학과
 {dshin, tskim, tgkwon@ce.cnu.ac.kr}

Software ATM Switch Server for ATM Adaptation Layer Type 2

Dong-Hoon Shin, , Tae -Suk Kim, Taeck-Geun Kwon
 Dept. of Computer Engineering, Chungnam National University

요 약

기존의 ATM (Asynchronous Transfer Mode) 스위치는 하나의 셀 안에 하나의 음성 데이터를 전송하는 방식이었다. 이러한 방식으로는 적은 양을 갖는 실시간 가변 비트율의 음성 데이터의 서비스를 제공하기에는 비효율적이었다. 이에 ATM 네트워크에서 사용자에게 실시간 가변 비트율로, 적은 양의 음성 데이터를 멀티플렉싱하여 하나의 ATM 셀의 유효부하에 실어 보내는 방식인 AAL2 (ATM Adaptation Layer Type 2) 프로토콜의 표준화가 이루어졌다. AAL2 스위치 서버는 AAL2 프로토콜로 인코딩된 ATM 셀을 스위칭하기 위해서 개발되었다. AAL2 스위치 서버는 CPS (Common Part Sublayer) 패킷을 추출해내는 부분과 CPS 패킷의 식별자를 갖고 식별자 스위칭하는 부분, 그리고 스위칭이 끝난 식별자를 갖는 CPS 패킷을 ATM 셀로 만들어 ATM 네트워크를 통해서 전송하는 부분으로 나뉜다. AAL2 스위치 서버를 3개의 부분으로 나누는 이유는 셀을 전송 받는 부분과, 식별자 스위칭하는 부분, 다시 셀을 전송하는 부분으로 나누어 병렬 처리를 함으로써 스위치의 성능을 향상 시킨다. 이 논문에서는 AAL2 스위치 서버를 구현하고, 그 활용방안에 대해서 알아보도록 한다.

1. 서 론

현재 급속히 증가하는 이동 통신 서비스의 확산을 수용하기 위하여 ATM 망 내부에서 실시간 가변 비트율 (rt-VBR : real-time variable bit rate) 데이터 서비스를 지원하는 ATM 상위 계층 프로토콜의 개발의 필요성이 고조되었다. 기존 ATM 프로토콜에 따르면 음성 데이터는 항등 비트율 (CBR : constant bit rate)로 AAL1 (ATM Adaptation Layer Type 1)[1]을 통해 전송된다. AAL1을 사용하여 이동통신을 위한 데이터 서비스를 제공할 경우에는 작은 크기의 음성 데이터를 위해 ATM 셀의 대부분을 비운 상태로 전송함으로써 대역폭의 낭비가 생긴다. 대역폭을 좀더 효율적으로 사용하고, 실시간 가변 비트율 데이터 서비스를 지원하기 위해서 새로운 프로토콜의 개발이 필요하게 되었다. 이러한 목적에서 1996년 I-TUT에서 AAL2 (ATM Adaptation Layer Type 2)[2][3][4] 프로토콜을 표준화하기 시작하였다. AAL2는 실시간 가변 비트율 (rt-VBR : real-time variable bit rate) 데이터 서비스를 위한 ATM 상위 계층 프로토콜이고, AAL2 프로토콜은 IMT-2000 서비스 통신망의 플랫폼이 ATM으로 채택됨에 따라서 더욱 관심이 고조되고 있다. 이에 본 논문에서는 실시간 가변 비트율 데이터 서비스를 제공하는 ATM 상위 계층인 AAL2 스위치 서버를 소프트웨어 기반으로 구현하고, 실험을 통해서 AAL2 스위치 서버의 성능을 측정해 본다.

2. AAL2 스위치 서버

AAL2 스위치 서버의 구성은 그림 1과 같다. 그림에서 볼 수 있듯이 AAL2 스위치 서버는 AAL2 방식으로 인코딩된 ATM 셀을 전송 받아 CPS 패킷을 추출해내는 역할을 하는 RCV 클래스, CPS 패킷의 식별자를 이용하여 스위칭을 하는 SWITCHING 클래스, 바뀐 식별자를 갖는 CPS 패킷을 다시 셀

로 바꾸는 역할을 하는 XMIT 클래스와 연결에 관한 스위치 자원의 분배와 스위치 테이블 관리를 위한 Controller 클래스로 구성된다.

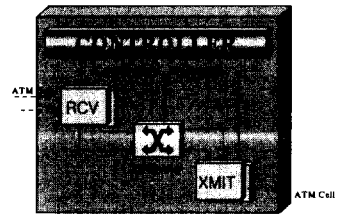


그림 1. AAL2 스위치 서버 구성

3. RCV 클래스

RCV 클래스는 ATM 셀을 전송 받아서 CPS 패킷을 추출하는 역할을 하는 클래스이다. RCV 클래스는 하나의 프로세스 형태로 실행이 되며, IDLE 이라는 하나의 상태로만 실행된다. 새로운 연결을 위한 VCC (Virtual Channel Connection)가 생성되면 이 VCC를 위한 RCV 클래스의 프로세스가 수행되어 ATM 셀을 전송받는다. RCV 클래스에 ATM 셀을 전송 받아 CPS 패킷을 추출해내는 과정은 그림 2를 보면서 자세히 살펴본다. 그림 2는 AAL2 스위치 서버의 RCV 클래스가 ATM 셀 3개를 전송 받아 CPS 패킷을 추출해내는 과정을 보인다. 우선, AAL2 연결이 생성되면, RCV 클래스는 하나의 프로세스 형태로 실행된다. RCV 프로세스는 우선 자신의 버퍼를 비워놓은 상태에서 ATM 셀의 도착을 기다리고 있다. 첫번째 ATM 셀에 CPS 패킷 1과 CPS 패킷 2의 일부가 실려 전송 되어왔다. RCV 프로세스는 자

신의 버퍼에 ATM 셀의 유료부하를 갖고 와서 CPS 패킷 1 을 추출해내고, 패킷 2 의 일부를 보관한다. 두 번째 ATM 셀이 자

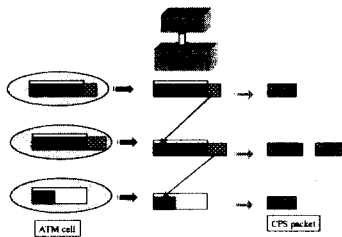


그림 3. RCV 클래스의 동작

신의 버퍼에 ATM 셀의 유료부하를 가지고 와서 CPS 패킷 1 을 추출해내고, 패킷 2 의 일부를 보관한다. 두 번째 ATM 셀이 도착하게 되면, CPS 패킷 2-1 과 2-2 을 합하여 CPS 패킷 2 을 추출해내고, 다시 셀내에 존재하는 패킷 3 을 추출한다. 마찬가지로 패킷 4 는 일부만 존재하므로 세번째 ATM 셀이 도착한 후에 패킷 4-1 과 4-2 를 합하여 패킷 4 를 추출한다. 위와 같은 방법을 되풀이 하면서, ATM 셀내에 존재하는 CSP 패킷을 추출해낸다. CPS 패킷을 추출해내는 과정에서 에러가 생기기도 한다. 이렇게 에러가 발생하게 되면 ATM 셀 안에 존재하는 모든 CPS 패킷을 사용할 수 없기 때문에 데이터 전송에 커다란 손실을 초래한다.

4. SWITCHING 클래스

스위칭 클래스는 하나의 프로세스 형태로 실행된다. 처음 프로세스가 실행되면, 스위칭 테이블을 초기화 하고, 테이블 내의 엔트리들을 모두 지운다. 스위칭 테이블은 아래 표에서도 알 수 있듯이 입력에 대한 <VPI, VCI, CID> 값과 출력에 대한 <VPI, VCI, CID> 값들의 쌍으로 이루어 진다.

표 1. 스위칭 테이블 엔트리

In_VPI	In_VCI	In_CID	Out_VPI	Out_VCI	Out_CID

스위칭 테이블은 제어기로부터 전달받은 연결에 관한 메시지들의 정보를 사용하여 연결을 설정 / 해지시 테이블 엔트리를 추가 / 삭제한다. 스위칭 테이블 엔트리는 RCV 클래스로부터 전송받은 CSP 패킷의 식별자를 스위칭하는데 사용한다.

5. XMIT 클래스

XMIT 클래스는 AAL2 스위치 서버에서 하나의 프로세스 형태로 실행된다. XMIT 프로세스는 RCV 프로세스가 하나의 상태에서만 실행되는데 반에 4 가지 상태를 전이 하면서 실행된다. XMIT 프로세스는 CPS 패킷을 ATM 에 실어보내는 역할을 한다. 그럼 XMIT 클래스가 갖는 4 가지 상태에 대해 자세히 살펴보자.

■ IDLE 상태

XMIT 클래스의 시작 상태이다. IDLE 상태는 ATM 셀을 구성하기 위한 버퍼가 비어 있는 상태이다. 이 상태에서는 새로운 CPS 패킷을 구성할 데이터를 전달받아 CPS 패킷 헤더를 추가하여 CPS 패킷을 완성한다. CPS 패킷 헤더의 구성은 CID 값과, 패킷 길이에 관한 정보, UUI 등의 정보를 구성하고, 여기에 HEC 값을 추가하여 완성한다. HEC 은 CRC5 를 사용한다. CRC5

를 일일이 계산하는 것은 많은 시간을 필요로 함으로, 본 연구에서는, CRC5 를 미리 계산하여, preCRC 라는 테이블 형태로 메모리에 저장하고, 이를 사용하여 좀 더 빠른 시간에 에러를 체크할 수 있도록 제공한다. preCRC 테이블은 CPS 헤더가 갖을 수 있는 값인 0x00000000 - 0xFFFFFFFF 의 CRC5 값을 미리 계산하여 테이블 형태로 만들어 놓고, 메모리에 로딩하여 사용한다. preCRC 테이블의 사용은 CPS 헤더 값에서 HEC 부분을 0 으로 놓고, 나머지 헤더의 값을 preCRC 테이블의 인덱스 값으로 사용한다. preCRC 테이블의 사용은, CPS 패킷을 구성할 때마다 수행해야하는 CRC5 의 계산의 시간을 테이블 참조 시간으로 줄일 수 있다. 완성된 CPS 패킷을 비어 있는 ATM 셀의 유료부하에 채워 넣고, 다음 상태로 전이한다.

■ PART 상태

PART 상태는 ATM 셀의 유료부하의 일부에 CPS 패킷이 저장되어 있고, 일부는 비어 있는 상태이다. ATM 셀의 비어 있는 부분을 채우기 위해서 새로이 전송할 데이터를 기다린다. 데이터를 전달 받으면 IDLE 상태와 같이 CPS 패킷을 구성하고, 다음 상태로 전이한다.

■ FULL 상태

FULL 상태에서는 하나의 ATM 셀을 완성하여 전송하고 난 후에 CPS 패킷 일부가 버퍼에 남아 있음으로 새로운 ATM 셀을 구성하여 버퍼에 남아 있는 CPS 패킷의 잔여분을 채워 넣는 역할을 한다. 이 상태에서 CPS 패킷의 잔여분의 길이 정보를 갖고 ATM 셀의 STF 필드에 ATM 셀내에 있을 첫번째 CPS 패킷의 헤더를 지시하는 포인터 정보를 설정한다. 이후 완성된 ATM 셀의 상태에 따라 다음 상태로의 전이가 일어난다.

■ SEND 상태

SEND 상태에서는 ATM 셀을 우선 전송시키고 XMIT 프로세스의 ATM 셀 버퍼를 비우고, IDLE 상태로 전이한다.

이상으로 XMIT 클래스의 4 가지 상태에 대해서 자세히 살펴 보았다. 이제 4 가지 상태들을 전이하면서 CPS 패킷을 ATM 셀로 만드는 과정을 그림 4 를 보면서 좀 더 자세히 살펴보자.

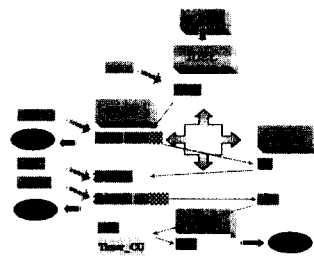


그림 4. XMIT 클래스의 동작

XMIT 프로세스는 IDLE 상태에서 실행된다. IDLE 상태에서 패킷 1 을 전달 받게 되면 ATM 셀 1 의 유료부하의 일부분에 채우고, ATM 셀 1 의 빈 부분이 있으므로 PART 상태로 전이한다. PART 상태에서 패킷 2 를 전달받아 ATM 셀 1 의 빈 유료부하 부분에 채운다. ATM 셀 1 의 빈 유료부하 부분이 작아서 패킷 2 일부인 패킷 2-1 만 채운다. 우선 완성된 ATM 셀 1 을 ATM 망을 통해 전송하고, FULL 상태로 전이한다. FULL 상태에서는 패킷 2 의 잔여분인 패킷 2-2 부분을 새로운 ATM 셀 2 에 채워 넣는다. ATM 셀 2 에 빈 부분이 있으므로 PART 로 전이시킨다. 마찬가지로 PART 상태에서 새로운 패킷 3 을 전달 받아서 ATM 셀 2 에 채워 넣었다. 이 후에도 ATM 셀 2 의 비어

있는 부분이 있으므로 다시 상태를 PART로 전이 시킨다. ATM 셀 3이 PART 상태에서 일정한 시간이 지난 후에도 데이터의 전달이 없으므로 상태는 SEND로 전이한다. SEND 상태에서 현 상태의 ATM 셀 3을 전송한다.

6. AAL2 스위치 제어기

AAL2 스위치 서버의 제어기는 AAL2 스위치 서버에게 연결을 설정 / 해지하기 위해 전송된 메시지의 정보를 이용하여 AAL2 스위치 서버의 자원을 관리한다. 연결에 관한 메시지는 새로운 연결을 요청하는 메시지와 이에 대한 응답 메시지, 현존하는 연결의 해지를 요청하는 메시지와 이에 관한 응답 메시지의 4가지 메시지를 사용한다. 새로운 연결을 설정할 경우에 제어기는 요청된 연결의 VPI, VCI 값이 현재 이미 연결되어 사용중인 VPI, VCI 이면 SWITCH 클래스의 스위칭 테이블 엔트리에만 새로운 연결을 위한 식별자를 기록하고, 요청된 VPI, VCI 값이 현재 스위칭 테이블 상에 존재하지 않는 in_VPI, n_VCI 이면 새로운 RCV 프로세스를, out_VPI, out_VCI 이면 XMIT 프로세스를 실행하여, 새로운 연결을 위한 서비스를 제공한다.

7. AAL2 스위치 서버 시뮬레이션

AAL2 스위치 서버의 시뮬레이션은 그림 5와 같은 연결을 갖고 실행이 된다. 이 실험을 위한 노드는 최소 4개 이상이 된다. 우선 스위치 서버에 AAL2로 인코딩된 ATM 셀을 만들어서 전송하는 TESTER 노드가 있고, AAL2 스위치 서버에게 연결 설정/해지 메시지를 보내는 MSG GEN 노드가 있고, 스위치 서버가 동작하는 노드가 있고, 스위칭된 ATM 셀을 받아서 CSP 패킷의 스위칭이 성공적으로 이루어졌는지를 관찰하는 MONITOR 노드가 있다. 본 연구에서는 AAL2 스위치 서버를 펜티엄 550 MHz 인 PC 2대에 리눅스를 깔아서 시뮬레이션을 실행해 보았다. 하나의 PC에는 AAL 스위치 서버의 프로세스들을 실행시키고, 다른 하나의 PC에는 TESTER, MONITOR, MSG GEN 노드를 실행하여 실험을 해보았다. 각 노드사이의 ATM 셀을 전송하기 위해서는 ATM NIC 카드를 이용하여 ATM law 셀을 전송하고, AAL2 스위치 서버내의 RCV, SWITCHING, XMIT 클래스사이의 CPS 패킷 전송은 IPC를 사용하였다.

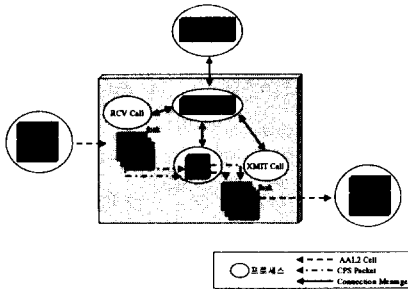


그림 5. AAL2 스위치 서버 시뮬레이션 모델

연결의 수행 과정을 보면, 우선 MSG GEN 노드에서 연결 설정 메시지를 AAL2 스위치 서버에게 보내고, AAL2 스위치 서버의 제어기가 연결 설정 메시지의 정보를 이용하여 연결의 위한 RCV, XMIT 프로세스를 실행시키고, SWITCHING 프로세스의 스위칭 테이블에 엔트리를 추가한다. 이제 AAL2 스위치 서버가 연결을 설정하고, 스위칭을 위한 준비를 마쳤다. TESTER 노드에서 AAL2로 인코딩된 ATM 셀을 AAL2 스위치 서버에

전송하고, MONITOR 노드에서 스위칭과 전송이 이루어졌는지 관찰한다. 또한, 새로운 연결을 추가하여 설정하기 위해서 MSG GEN 노드에서 연결 설정 메시지를 AAL2 스위치 서버에 전송하여 새로운 연결을 설정하고, 새로운 연결을 위해서 TESTER 노드와 MONITOR 노드에서 AAL2로 인코딩된 ATM 셀을 전송하면서 관찰해본다. 연결의 해지 방법은 MSG GEN 노드에서 연결 해지 메시지를 AAL2 스위치 서버에 전송하고, AAL2 스위치 서버 제어기는 연결 해지 메시지의 정보를 이용하여 RCV, XMIT 프로세스를 종료하고, SWITCHING 클래스의 스위칭 테이블 엔트리에서 연결에 관한 정보를 제거함으로써 연결을 해지를 수행한다.

8. 시뮬레이션 결과

AAL2 스위치 서버의 성능은 노드간의 ATM 셀 전송속도와 AAL2 스위치 서버 클래스 사이에 CPS 패킷을 전송하는 IPC 속도에 영향을 받는다. 본 연구에서는 각 노드들의 전송속도와 IPC의 전송속도 지연을 최소화하여 성능을 측정해 보았다. 즉, AAL2 스위치 서버의 핵심 부분인 RCV, SWITCHING, XMIT 클래스를 하나의 프로세스에 실행하여 성능을 측정해 본 것이다. 이 경우 CPS 패킷 전송은 큐를 통해서 이루어진다. 그림 7(A)는 AAL2 스위치 서버의 성능을 측정한 그래프이고, 그림 7(B)는 AAL2 스위치 서버의 핵심 부분만의 성능을 측정한 것이다. 그림에서도 알 수 있듯이, AAL2 스위치 서버의 핵심 성능은 우수하나 ATM 셀과 CPS 패킷의 전송으로 인한 지연으로 인해 AAL2 스위치 서버 전체의 성능의 저하가 발생하였다.

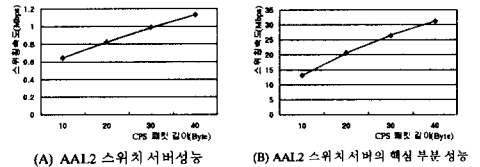


그림 7. AAL2 스위치 서버의 성능

9. 결론 및 향후 연구과제

본 논문에서는 실시간 가변 비트율 데이터 서비스를 지원하기 위한 AAL2 스위치 서버를 소프트웨어로 구현하고 그 성능을 측정해 보았다. 이러한 연구는 AAL2 스위치 서버는 향후 IMT-2000 서비스 통신망에서 사용될 수 있다는데 의의가 있다. 향후 연구 과제는 AAL2 스위치 서버의 스위칭 속도가 만족할만한 수준까지 도달하지 못 하였으므로, AAL2 스위치 서버의 성능 향상을 위해서 ATM 셀의 전송, CPS 패킷 전송, CRC5 체크 등을 하드웨어를 통해 실행하는 연구가 더 진행될 수 있을 것이다.

참 고 문 헌

- [1] ITU-T I.363.1, "B-ISDN ATM Adaptation layer Specification : Type 1 AAL", 1996.
- [2] ITU-T I.363.2, "B-ISDN ATM Adaptation layer Specification : Type 2 AAL", 1997.
- [3] H.K. Lim, "AAL2 - A New ATM Adaptation Layer for Low Bit Rate Speech and Data : Issues and Challenges", 1998.
- [4] Behram.H and John.H, "AAL2 - A New ATM Adaptation Layer for Small Packet Encapsulation and Multiplexing", Bell Lab Journal, Spring, 1997