

VPI/VCI Look-up Table 구조의 구현 방식에 따른 성능 비교

정회원 김진수*

Architecture of VPI/VCI Look-up Table and Its Performance Comparison for Hardware Implementation Methods

Jin-soo Kim* *Regular Member*

요 약

ATM 망은 CO (connection oriented) 망으로서 ATM 셀 헤더의 VPI/VCI로 제공되는 가상 주소 식별자 정보를 이용하여 연결 설정을 관리한다. 최근에 다양한 형태로 구현되어 ATM/ADSL 단말기가 사용자들에게 제공되고 있다. 사용자 단말기가 수용할 수 있는 연결 접속의 수는 VPI/VCI 룩업 테이블의 크기 및 구조로 결정되며, 개개의 연결 접속 정보를 고속으로 처리하고 관리하는 것은 SAR (Segmentation and Reassembly) 기능에 의한 서비스 품질과 단말기 가격을 결정하는데 매우 중요한 역할을 한다. 본 논문에서는 VPI/VCI 룩업 테이블의 구현 가능한 구조를 제시하고, 주소 식별 설계 방식에 따른 성능 변화를 연결 접속 정보 접근에 대한 평균 읽기 사이클 횟수 측면에서 비교 분석한다. 또한, 연결 설정을 각 서비스의 PCR (peak cell rate)에 따라 VPI/VCI 룩업 테이블의 내용을 정렬하면, 설정한 모의 실험 환경에서 단순 정렬 방식에 비해 19%~34% 이상의 성능 개선이 이루어짐을 모의 실험을 통하여 보인다. 이 결과를 확장하여, PCR 크기 순서 정렬 및 로그 탐색이 가능한 경우에는 더욱 성능이 개선됨을 보이고, 성능 개선에 대한 상한 (upper bound)과 하한 (lower bound)에 대해 논하고 모의실험을 통하여 확인한다.

ABSTRACT

In ATM networks, call setup and management are dealt with VPI/VCI fields of the ATM (Asynchronous Transfer Mode) cell header. The number of acceptable connections in ATM terminal equipment or adaptor is dependent on the size and structure of VPI/VCI Look-up Table at SAR device. Accordingly, when the SAR device is implemented on a hardware product, it is very important to control, in a high speed, the connection informations including segmentation/reassembly informations corresponding to each sending/receiving ATM cells. First, this paper describes several structures of VPI/VCI Look-up Table and their performance comparisons are analyzed in terms of the average number of read cycles necessary to access to the VPI/VCI Look-up Table. Then, through computer simulations, it is shown that a VPI/VCI Look-up Table with PCR (Peak Cell Rate) order arrangement can be improved about up to 19%~34%, compared to a simple arrangement method under the experimental conditions of this paper. Additionally, by extending this method, this paper discusses the method of VPI/VCI Look-up Table with the PCR order arrangement/logarithmic search and then the upper/lower bounds of its performance.

* 국립 한밭대학교 정보통신 컴퓨터공학부(jskim67@mtut.ac.kr)

논문번호 : 00382-1002, 접수일자 : 2000년 10월 2일

* 본 연구는 2000년도 국립 한밭대학교 교내학술 연구비 지원으로 수행되었음.

I. 서론

ATM 망은 사용자 정보에 대한 연결 수락 제어가 수행되며, 연결 설정이 허용된 후에는 사용자 정보를 주고받게 된다^[1]. 그림 1은 SAR (Segmentation and Reassembly) 디바이스의 사용자 정보 송수신에 따른 하드웨어 내부 구조를 개념적 블록도로 나타낸 것이다. 사용자 정보는 53바이트 크기의 ATM 셀의 패킷으로 분해되어 송신되며, 수신측에서는 도착하는 ATM 셀 헤더에 있는 VPI (Virtual Path Identifier)/VCI (Virtual Channel Identifier)를 이용하여 재조립 (reassembly)을 수행한 후에 사용자 계층으로 그 정보를 인터럽트 형식으로 알려준다^{[1][2][3]}. 만일, 수신되는 AAL-5 CPCS (ATM Adaptation Layer-5 Common Part Convergence Sublayer) 패킷의 크기가 48바이트보다 큰 경우에는 두 개 이상의 ATM 셀로 나누어 수신이 되며, 재조립을 위해 메모리의 특정위치에 저장된다. 이때, 재조립을 행하는 호스트는 저장된 데이터에 대한 저장 위치, 수신된 정보의 길이, CRC-32 정보 그리고 버퍼 충만도 정보 등을 효과적으로 관리 유지하여야 한다. 특히, 단말기에서 연결 설정된 접속 (connection) 개수를 많이 수용하고자 하는 경우에는 개개의 연결 접속에 대한 정보와 분할 및 재조립에 대한 정보를 고속으로 발체하는 것이 필수적이다. 만일, 그렇지 못한 경우가 발생한다면, 개개의 연결에 해당하는 서비스 품질 (QoS : Quality of Service)에 다소 악영향을 줄 수가 있다.

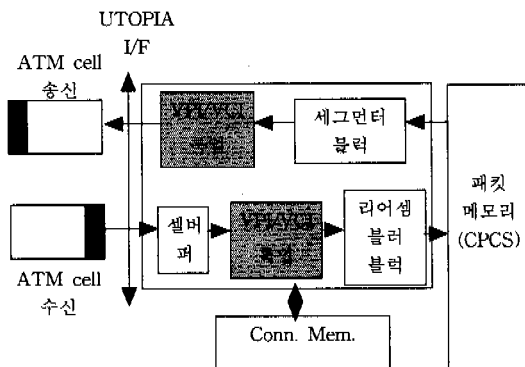


그림 1. 데이터 송수신에 대한 SAR 블록의 개념적 블록도

최근에는 다양한 형태로 ATM 디바이스가 개발되어 여러 사용자들에게 보급되고 있다^{[4][5][6]}. ATM NIC (Network Interface Card) 또는 stand-alone형

태로 제공되는 SAR(Segmentation and Reassembly) 디바이스는 응용 환경에 따라 값이 고가이거나 또는 저렴한 하드웨어로 구축되고 있다. 만일, 값이 저렴한 형태로 많은 개수의 연결을 설정할 경우에는 SAR 기능의 원활한 수행과 비용 측면에 있어서 많은 제약이 따른다. 예를 들어, ATM over DSL과 같이 저속의 프로세서를 사용하고, 메모리 용량 또한 제한적인 경우에 연결 접속 정보를 효과적으로 관리하지 못한다면, 사용자가 요구한 서비스 품질을 만족시킬 수 없는 경우가 발생할 수도 있다. 그러나, 메모리를 충분히 확장할 수 있거나, 고속의 프로세서를 사용할 수 있는 고가의 ATM-LAN 단말기의 경우에는 다양한 융통성을 확보하면서 하드웨어를 설계하는 것이 바람직하다.

그렇다면 어떤 방식으로 연결 접속을 관리해야 하는가 하는 물음이 생긴다. 기존의 대표적인 기법으로는 크게 두 가지로 나눌 수 있다^{[4][5][6]}. 첫째, 연결 설정되는 순서대로 메모리의 특정 영역에 VPI/VCI 값을 기억하며, 연결 설정 순서에 따라 연결 번호 또는 채널 번호를 기입하는 방법이다. 저가로 수용하는 가상 연결의 수가 적은 경우에 사용될 수 있는 방법이다. 이 방법은 수신측에 도착하는 ATM 셀의 서비스 속도가 매우 빠른 경우에는 서비스 지연이 발생할 수 있다. 이런 문제가 발생되지 않기 위해서는 고속의 프로세서 사용이 필수적이다. 둘째, ATM 셀 헤더에 있는 VPI/VCI 비트에 대해 마스크를 취하는 방법이다. 이 경우에는 ATM 셀 헤더로 표현 가능한 방법의 수가 1천6백만 가지 이상이다. 따라서, 이를 모두 표현하기 위해서는 매우 많은 양의 메모리를 요구하게 된다(특정 PVC 채널만을 설정하는 경우에는 문제가 되지 않음). 충분한 양의 메모리를 사용하지 못하는 경우에는 마스크에 의해 동일한 결과를 얻는 경우가 발생하게 되어 잘못된 연산을 수행하여 오동작을 일으킬 가능성이 존재한다. 셋째, 위에서 제시한 두 가지 방법을 혼용하여 사용하는 것이다. 이 경우에는 SAR 기능을 저가의 형태로 최적화되지 못한다는 단점이 있다. 그러나, 일반적인 범용의 칩에는 사용자에게 의해 선택 가능하도록 해두고 있다. 마지막으로 VPI/VCI 값에 대한 연결 정보를 두 개의 사이클만에 읽어내도록 하는 방법을 사용한다. 이 경우에는 CAM (Content Addressable Memory)를 사용하기 때문에 실제 단순 VPI/VCI 정렬 방식에 비해 전력 소모량이 두 배 이상이며, 칩 면적 또한 1.5 배이상 소요된다. SAR 기능은 다양한 응용 환경에 사용되고,

이에 따라 다양한 형태로 설계되고 이용될 수 있다. 이런 관점에서 본 논문에서는 연결 접속 정보의 효과적 접근을 위한 VPI/VCI 룩업 테이블의 구조와 각 구조에 따라 개개의 연결 정보를 관리 유지할 수 있는 방식에 대한 성능 분석을 수행한다.

II. VPI/VCI 룩업 테이블의 구조와 설계 방식

ATM 단말기에서 수행하는 연결 접속에 대한 관리는 ATM 셀 헤더에 있는 VPI/VCI 값을 연결 접속 정보를 저장하는 메모리의 특정 위치에 연계시켜 관리함으로써 가능하다. UNI (User Network Interface)에서의 ATM 셀 헤더에서 제공되는 가상 주소 식별자인 VPI (virtual path identifier)는 8비트이고, VCI (virtual channel identifier)는 16비트이다. 모든 가능한 가상 주소 식별자를 구별하기 위해서는 2^{24} 만큼의 Table 크기가 필요하다. 만일, 단일 VPI/VCI 룩업 테이블의 크기가 4 bytes(32 bits)가 필요하다면, 이를 위해 소요되는 메모리의 크기는 67 Mbytes만큼의 메모리가 필요하게 된다. 따라서, 모든 가능한 연결 접속 조건을 구현하는 것은 거의 불가능할 뿐만 아니라, 구현하는 것 자체가 낭비이다. 본 장에서는 실제 제품으로 구현되고 있거나, 또는 구현 가능한 VPI/VCI 룩업 테이블의 설계 방법에 대하여 논한다.

1. ATM 셀 헤더의 VPI/VCI 필드의 마스크를 이용한 주소 식별 설계 방법

가장 손쉽게 설계할 수 있는 연결 정보 관리 방법이 VPI/VCI 비트들 중의 특정 부분을 마스크를 취하여 그 결과로 얻어지는 정보를 이용하여 VPI/VCI 룩업 테이블에 대한 주소를 발생시키고, 연결 접속 메모리 (connection memory)에 대한 기본 주소 (base address)를 얻는 방법이다. 구현하는 가장 흔한 방법으로는 VPI 별로 특정 위치의 VCI 마스크 값을 다르게 할당하고, 이 값을 VCI 필드의 특정 MSB (Most Significant Bit) 이상에 대해 마스크를 취하여 얻어지는 연결 접속 메모리의 위치를 얻게 된다.

이 방법은 수신되는 ATM 셀 헤더의 값을 이용하여 제어 클럭 (control clock)의 2 사이클 내에 연결 번호 (connection number)를 가져올 수 있기 때문에 처리 지연이 매우 작은 장점이 있다. 그러나, 많은 양의 메모리 사용에 따른 비용적인 측면에서 바람직하지 못하며, 단말기가 수동적으로 제어되

므로 인해 사용되지 않는 메모리 영역이 존재하며, 따라서, 메모리의 사용을 극대화시킬 수 없는 단점이 대두될 수 있다. 즉, 특정 VPI/VCI 필드 중에 특정 부분이 사용되지 않는다면, 그 만큼 메모리의 낭비를 초래하게 되는 것이다. 또한, 특정 VPI에 대해 VCI 가상 연결 접속 수에 대한 제약이 따를 수가 있다. 예를 들면, VCI의 LSB (Least Significant Bit) 4 비트만을 이용하여 연결 정보를 찾는다고 할 경우에는 VCI=1은 VCI=17과 동일한 위치의 메모리를 사용하게 될 것이므로, 특정 VCI에 대해서는 연결 설정이 더 이상 불가능한 현상이 일어날 수 있다. 이 방법은 사실상 고가의 ATM 단말기 또는 스위치를 구현하는 방법으로 많이 사용되고 있으며, 또한 널리 알려진 방법이다.

2. VPI/VCI 값의 단순 정렬을 이용한 주소 식별 방법

VPI/VCI 필드 값을 마스크를 취하여 연결 번호를 얻어 주소 식별을 결정하는 방법은 단순하며 구현하기 간단하다. 그러나, 이와 같은 방법은 메모리가 작은 경우에 쓰기에는 불편하고, 특히 망에서 연결 정보를 미리 결정한 경우인 PVC (Permanent Virtual Connection)를 수용하고자 하는 환경에 적용하기에는 상당한 문제점을 포함할 수 있다. 따라서, 이와 같은 환경을 회피하기 위해서 생각해 볼 수 있는 방법으로서 VPI/VCI 값을 연결 순서에 따라 정렬을 시킨 다음, 특정 연결 정보를 찾아가는 방법이다. 만일, 기존의 가상 연결 접속이 해제될 경우에는 그 필드에는 플래그 필드를 두어 무효 (invalid)로 표현하고, 새로운 연결 접속이 들어오면, 그 곳에 새로운 연결 접속 정보를 기억시킨 후에 그 필드에는 유효 (valid)로 표현한다.

이 방법은 메모리가 작은 경우에 효과적으로 사용될 수 있으며, 따라서 저가의 단말기를 구현하는데 직접 사용되고 있는 방법이다⁶⁾. 그러나, 이 방법의 제한점은 ATM 셀이 분할(segmentation)되거나 또는 재조립(reassembly)되는 경우에는 항상 일정 순서에 의해 메모리를 찾아 들어가야 한다는 것이다. 또한, 각 연결 접속별로 셀 지연 변이 (cell delay variation)이 다르게 나타나고, 우연히 PCR (Peak Cell Rate)이 아주 큰 서비스에 대한 연결 정보가 제일 많은 읽기 사이클을 필요로 하여 읽혀진다면 셀 지연 변이를 크게 겪을 수 있는 단점이 있다. 저가의 UBR (Unspecified Bit Rate Service) 서비스를 주로 고려한 응용 환경인 ATM over

DSL (Digital Subscriber Line)의 경우에는 다소 무시될 수도 있으나, 경우에 따라서는 서비스의 품질에 영향을 크게 줄 가능성이 내재해 있다.

3. PCR (Peak Cell Rate) 크기 순으로 정렬을 이용한 주소 식별 방법

VPI/VCI 값을 이용하여 주소 식별 정보를 구하는 경우에는 연결 관리를 위한 메모리의 사용을 적게 할 수 있는 장점이 있으나, 위의 예시에서처럼 PCR의 값 변동에 따른 연결 정보에 따라 메모리에 읽기 쓰기 부담은 달라진다. 만일, 연결 설정된 가상 연결의 개수가 증가할 경우에는 나중에 설정된 연결은 메모리에서 데이터를 읽고 쓰는데 부담을 갖게 된다.

이와 같은 문제점을 극복하기 위한 근본적인 해결 방안은 메모리의 접근이 많은 연결 설정일수록 VPI/VCI 록업 테이블에서 낮은 주소의 값에 위치시키고, 반대로 메모리의 접근이 적은 연결 설정은 가능하면 VPI/VCI 록업 테이블의 높은 곳에 위치시키는 방법이다. 이와 같이 정렬을 시키면, 높은 비트율을 요구하는 서비스에 대한 가상 연결은 연결 정보를 얻는데 지연이 크게 소모되지 않게 된다. 본 논문에서는 PCR의 값이 큰 경우에 따라 낮은 위치에 위치시키도록 하는 방법을 제안한다.

새로운 접속이 설정될 때마다 그 연결의 PCR 크기에 따라 위치를 다시 정렬시키면 이 방식을 쉽게 수용할 수 있다. 따라서, 이 방식은 새로운 연결 접속이 설정되거나 또는 해제되는 경우에는 VPI/VCI 록업 테이블을 호스트가 수정해주어야 하는 부담이 있을 수 있다. 그러나, 일반적으로 호 설정이나 새로운 가상 연결의 설정 처리 시간은 셀 단위의 처리 시간 또는 수신 시간 간격에 비해 매우 긴 시간 영역이기 때문에 CPU의 연산 부담은 크지 않다^{4,5)}.

III. VPI/VCI 록업 테이블의 구조에 따른 성능

1. VPI/VCI값의 단순 정렬을 이용한 주소 식별 방법의 평균 메모리 읽기 사이클 수

연결 설정된 순서에 따라 VPI/VCI 값을 특정 연결 정보에 위치시키는 방식에 대하여 평균적으로 VPI/VCI 록업 테이블을 읽는데 소요되는 메모리 읽기 사이클 수는 다음과 같이 표현된다.

$$\text{평균 읽기 사이클 수} = \frac{\sum_{i=1}^N i}{N} \quad (1)$$

여기서 N 은 주어진 특정 순간의 설정된 가상 VC 연결의 개수이다. 따라서, 단순한 순서로 VPI/VCI 록업 테이블을 읽게 됨에 따라 평균적으로 읽어 들여야 할 소요 시간은 크며, 연결 설정되는 VC의 수에 따라 매우 많은 지연을 요구하게 된다. 이 방법은 메모리가 작은 경우에 사용될 수 있는 방법 중의 한 가지이다.

2. PCR (Peak Cell Rate) 크기 순 정렬을 이용한 주소 식별 방법의 평균 메모리 읽기 사이클 수

PCR의 크기 순으로 VPI/VCI 값을 정렬시켜 놓는 경우에는 순서적으로 읽기를 할 경우에는 다음과 같은 평균 읽기 사이클 수가 얻어진다.

$$\text{평균 읽기 사이클 수} = \frac{\sum_{i=1}^N i \times r_i}{\sum_{i=1}^N r_i} \quad (2)$$

여기서 $r_i, i=1, \dots, N$ 은 각 순서로 저장된 VPI/VCI에 대한 PCR 값을 나타내고, N 은 특정 순간의 설정된 가상 VC 연결 접속 수를 나타낸다. 이 식에서 알 수 있듯이 평균 읽기 사이클 수는 가상 VC 연결 접속에 대한 PCR의 크기에 직접적으로 영향을 받게 된다.

3. VPI/VCI 값에 따른 정렬과 로그 탐색 방법에 의한 평균 메모리 읽기 사이클 수

VPI/VCI 록업 테이블에 특정 연결 정보를 등록하는 것은 호 설정 및 연결 수락 제어가 끝난 후에 한차례 이루어진다. 이때, 하드웨어 스스로 메모리 영역에 연결 설정 정보를 기억시킬 수도 있으나, ARM (Advanced RISC Machines)과 같이 특정 CPU가 VPI/VCI 록업 테이블을 관리할 수도 있다. 이 경우에는 VPI/VCI 록업 테이블을 VPI/VCI 값에 따라 손쉽게 정렬할 수 있으며, 특정 연결 정보를 얻기 위해서는 로그 탐색을 수행하여 찾을 수도 있다. 이와 같은 경우에는 평균 읽기 사이클 수는 크게 개선될 수 있다. 즉,

평균 읽기 사이클 수 =

$$\frac{1}{N} \left[\sum_{i=1}^{\lceil \log_2 N \rceil} i \times 2^{i-1} + (\lceil \log_2 N \rceil + 1) \times (N - 2^{\lceil \log_2 N \rceil}) \right] \quad (3)$$

이 된다. 여기서, $\lceil \log_2 N \rceil$ 은 $\log_2 N$ 의 정수부만을 취한 값이고, N 은 특정 순간에 설정된 가상 VC 연

결 수이다.

만일, 로그 탐색이 가능하고, 또한 PCR 비율로 접근되도록 조절이 가능하게 되면, 설정된 연결에 대해 읽기 사이클 수를 크게 감소시킬 수 있게된다. 따라서, 이와 같은 경우에는 (2)식과 (3)식을 이용하여 전개하면

$$\begin{aligned} \text{평균 읽기 사이클 수} = & \frac{1}{\sum_{j=1}^N r_j} \left[\sum_{k=1}^{\lceil \log_2^N \rceil} k \cdot \left(\sum_{j=2^{k-1}}^{2^k-1} r_j \right) + \right. \\ & \left. (\lceil \log_2^N \rceil + 1) \times (r_{2^{\lceil \log_2^N \rceil}} + r_{2^{\lceil \log_2^N \rceil} + 1} + \dots + r_N) \right] \end{aligned} \quad (4)$$

을 얻게된다. 여기서, $r_1 \geq r_2 \geq \dots \geq r_N$ 로 정렬된 경우이다. 이것은 로그 탐색에 대해 최소의 접근을 요구하는 경우가 되며, (4)식은 (3)식에 대한 하한(lower bound)으로서의 의미를 갖게 된다. 그러나, 만일, (4)식에서 $r_1 \leq r_2 \leq \dots \leq r_N$ 이 되도록 정렬이 된다면, 이것은 로그 탐색에서 최대로 많은 접근을 요구하는 경우가 된다.

평균 읽기 사이클 수 =

$$\begin{aligned} \frac{1}{\sum_{j=1}^N r_j} \left[\sum_{k=1}^{\lceil \log_2^N \rceil} k \cdot \left(\sum_{j=2^{k-1}}^{2^k-1} r_{N-j+1} \right) + \right. \\ \left. (\lceil \log_2^N \rceil + 1) \times (r_{N-2^{\lceil \log_2^N \rceil} + 1} + r_{N-2^{\lceil \log_2^N \rceil}} + \dots + r_1) \right] \end{aligned} \quad (5)$$

위의 식은 (4)식과 같이 $r_1 \geq r_2 \geq \dots \geq r_N$ 로 정렬된 경우이다. 따라서, 위의 (5)식은 (3)식에 대한 상한(upper bound)으로서 의미를 갖게 되고, 다음과 같이 얻어진다.

IV. 모의실험 결과와 분석

1. 모의실험 환경

그림 2는 본 논문에서 사용된 모의 실험에 대한 개념적인 구성도를 나타낸다. VPI/VCI 룩업 테이블에 접근하는 수(읽기 사이클 수)는 PCR 크기에 비례하기 때문에, ATM 셀은 단위 시간당 일정률로 발생되도록 한다. 물론, 실제로 발생하는 트래픽 특성은 연결 접속된 서비스의 특성에 따라 매우 다른 형태로 발생되지만, 본 논문에서는 분석의 효율을 위해 PCR 크기의 일정 비트율(CBR: constant bit

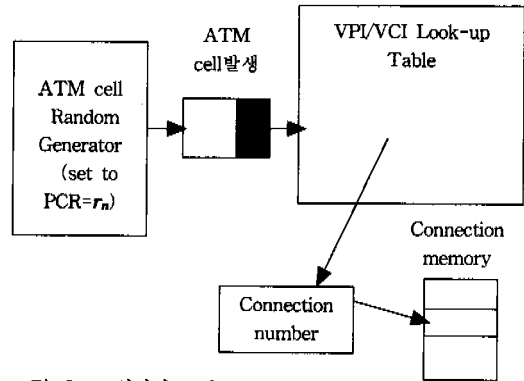


그림 2. 모의실험 구성도

rate)로 발생한다고 가정한다(이와 같은 가정은 본 논문 전개상에 일반성을 잃지 않는다). 본 논문에서는 모의 실험을 위하여 임의의 i번째 연결에 대한 PCR, $r_i = \frac{10 \cdot i}{N} Mbps, i=1, \dots, N$ (N: 가상 연결 수)로 설정하고, n번째 연결 접속에 대한 PCR인 r_n 은 r_i 중에서 임의의 값을 랜덤하게 선택하도록 한다. 이때 다른 연결 접속과 동일한 r_i 을 가질 수 있도록 한다. 본 논문에서는 모의실험 결과로 인한 특성이 충분히 반영되도록 하기 위해 10분 동안의 트래픽 발생 시간을 설정하여 모의실험을 수행한다. 여기서, 식(2)-(5)에 의해 쉽게 유추할 수 있듯이, r_i 값의 선택 방식에 따라 성능 차이는 다소 다르게 나타날 수 있으므로, VPI/VCI 룩업 테이블 설계 방식에 대한 상대적 성능 차이 비교에 중점을 두고자 한다.

2. 모의실험 결과

모의 실험은 두가지 형태로 나누어 진행된다. 우선, 저가형 ATM 단말기를 ADSL (G.dmt/G.lite) 단말기와 같이 가상 VC 연결이 64개 이하를 수용할 수 있는 것으로 가정한다. 그리고, ATM-LAN 또는 ATM 단말기 등과 같이 64개 이상의 연결 접속을 수용할 수 있는 것을 고가형 ATM 단말기로 고려한다. 그림 3은 저가형 ATM 단말기에서 PCR 정렬 방식과 단순 정렬 방식에 의한 성능 차이를 보여준다. 4개의 가상 연결 접속만을 수용한 경우에는 약 19% 정도의 성능 개선이 있으며, 64개의 연결 접속을 수용하였을 경우에는 약 31%의 성능이 개선됨을 알 수 있다. 모의실험 특성과 식 (2)에 의한 결과는 거의 동일한 특성을 보여 주고 있다.

그림 4는 고가형 ATM 단말기에서 PCR 정렬 방식과 단순 정렬 방식에 의한 성능 차이를 보여준다.

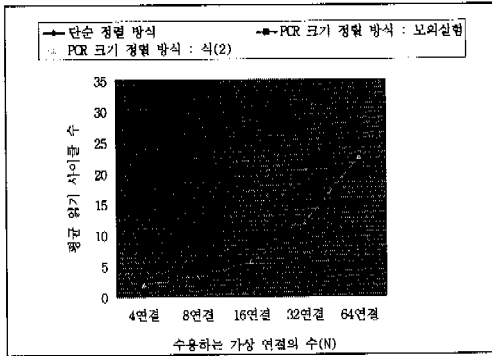


그림 3. 저가형 ATM 단말기 모델에 대한 모의실험 결과 (수용하는 가상 연결 수≤64).

128개 연결 설정에서는 32% 정도의 성능 개선이 이루어지며, 2048개 연결 설정에서는 33% 정도의 성능 개선이 이루어진다. 즉, 연결 접속의 개수가 많아질수록 성능 개선은 더욱 확대됨을 확인할 수 있으나, 저가형 ATM 단말기 모델에 비해 연결 접속 수가 지수 함수적으로 증가하더라도 크게 증가되지 않음을 확인할 수 있다. 실제로 본 논문에서 제시한 PCR 정렬 방식은 구현의 복잡도 등을 고려할 때 저가형 ATM 단말기에 적절할 것이다. 그리고 실제 고가형 ATM 단말기에서는 많은 양의 메모리를 사용하여 마스킹 방식으로 구현하고 있는 실정이다. II장에서 기술한 것처럼 고속 및 다기능의 MCU 칩을 이용한 ATM 단말기는 다양한 방식으로 VPI/VCI 룩업 테이블을 구현하는 것이 가능하지만, 본 논문에서 제시한 로그 탐색을 쉽게 수행하도록 설계하는 것 또한 가능하다. 이 경우에는 성능 개선을 크게 개선시킬 수가 있을 것이다. 그림 5는 저가형 ATM 단말기에 MCU가 장착되어 VPI/VCI 룩업 테이블이 관리될 경우에 얻어지는 성능 비교이다.

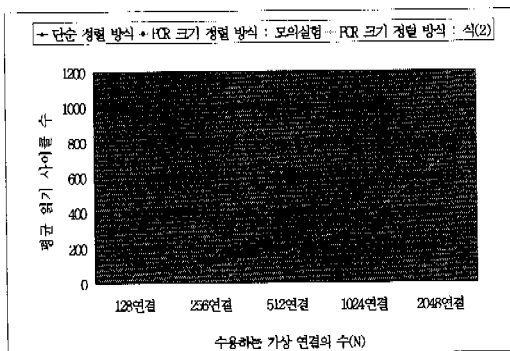


그림 4. 고가형 ATM 단말기 모델에 대한 모의실험 결과 (수용하는 가상 연결 수≥128).

우선, 'PCR정렬+Log탐색'이라고 한 것은 VPI/VCI 룩업 테이블의 내용을 PCR 크기 별로 정렬하고, 로그 탐색 방식을 MCU가 수행하는 방식을 말한다. 이 경우에 대해 성능이 최대로 개선될 수 있는 경우('BestCase'로 표시하며, 식(4)에 의해 결정되는 양임.)와 오히려 로그 탐색 방식이 최대로 나쁜 경우 ('WorstCase'로 표시하며, 식 (5)에 의해 결정되는 양임.)에 대해 각각 그 결과를 도시하였다. 그림 6은 고가형 ATM 단말기 모델에 대해서도 동일한 방식으로 모의 실험을 하여 얻은 결과를 나타내었다. 그림 3과 4에 비해 월등히 성능 개선이 크게 이루어지는 것을 확인할 수 있다. 이와 같은 방식은 특수한 응용 환경에 사용될 수 있을 것이다. 한편, 그림 5와 6에서도 모의실험에 의한 결과와 식 (3)에 의한 결과는 주어진 비트율이 결정되면 거의 일치하는 결과는 낳고 있음이 관찰된다.

V. 결론

본 논문에서는 하드웨어 구현을 기초로 VPI/VCI 룩업 테이블의 구현 방법과 각 구성 방식에 대한 장단점을 논하고, 이를 개선하기 위한 방안을 제안하였다. 즉, 고가 및 저가 형태의 다양한 구조를 갖는 ATM 단말기가 소비자에 의해 요구됨에 따라 VPI/VCI 룩업 방식을 고속으로 처리하기 위해 PCR 크기순 정렬 방식을 제안하였다. 또한, 고속 MCU 프로세서를 사용할 수 있는 경우에 대해 성능 개선 방안을 제안하였고, 더불어 성능 개선 정도를 확인하기 위한 상한과 하한을 각각 비교 분석하였다. 본 논문에서 설정한 모의실험 환경에서 제안한 PCR 정렬 방식에 의한 VPI/VCI 룩업 테이블의 구성은 단순 정렬 방식에 비해 19%~34%의 성능

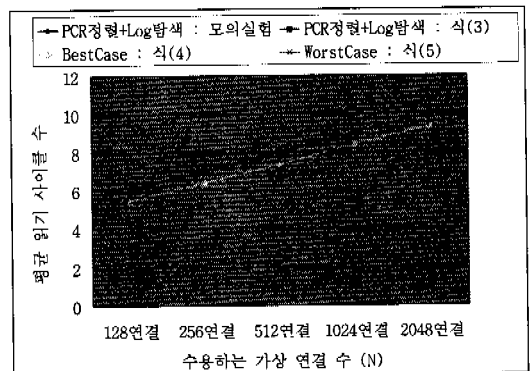


그림 5. 저가형 ATM 단말기의 VPI/VCI 룩업 테이블에 대해 로그 탐색이 가능한 경우.

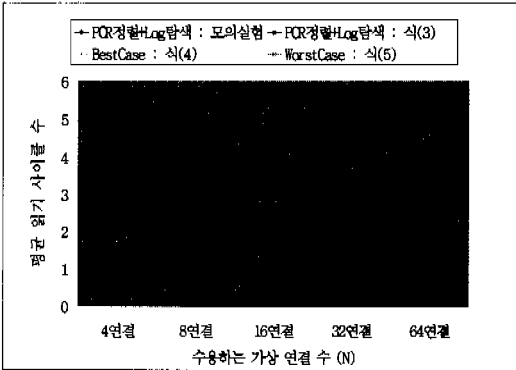


그림 6. 고가형 ATM 단말기의 VPI/VCI 룩업 테이블에 대해 로그 탐색이 가능한 경우.

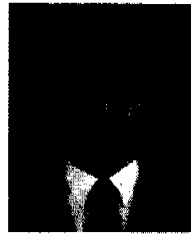
개선을 이루었으며, 고속 MCU가 사용 가능한 경우에는 매우 뛰어난 성능 개선을 확인하였다. 결론적으로 본 논문에서는 메모리의 사용이 제한적인 경우 즉, 전력 소모가 크고, 칩의 면적이 어느 정도 제한적이어서 많은 양의 메모리 사용이 어려운 경우에 효과적으로 사용될 수 있을 것이다. 앞으로 다양한 연결 접속에 대한 트래픽 파라미터를 가변적으로 설정한 경우에 대해 복합적으로 연구 분석이 진행될 필요가 있다.

참 고 문 헌

- [1] ATM Forum Draft Specification, "Traffic Management Specification Version 4.1," March 1999.
- [2] ITU-T Draft G.999.2, "Splitterless Asymmetric Digital Subscriber Line (ADSL) Transceivers," 12 October 1998.
- [3] IUT-T I.363.2, "B-ISDN ATM Adaptation Layer Specification - Protocol Layer Requirements," September 1997.
- [4] Motorola Users Manual "MPC860 SAR," September 1999.
- [5] Samsung Users Manual "KS8946 SAR," December 1999.
- [6] Fujitsu Users Manual "FireStream155 MB86697A," April 2000.
- [7] G.Kesidis, "ATM Network Performance," Kluwer Academic Publishers, 1996.

김진수(Jin-soo Kim)

정회원



1991년 2월 : 경북대학교 전자공학과 (공학사)

1993년 2월 : 한국과학기술원 전기및전자공학과 (공학석사)

1998년 8월 : 한국과학기술원 전기및전자공학과 (공학박사)

1995년 5월~2000년 3월 : 삼성전자 선임연구원

2000년 4월~현재 : 국립 한밭대학교 정보통신 컴퓨터 공학부 조교수

<주관심 분야> 멀티미디어 다중화및통신, 패킷 비디오, ATM 트래픽 제어