

# 홈 게이트웨이용 칩셋 및 스위치 구조 설계

최광순<sup>0</sup> 정광모  
전자부품연구원 고속네트워크연구센터  
(lenon, jungkm)<sup>0</sup>@keti.re.kr

## Korea Electronics Technology Institute

Kwang-Soon Choi<sup>0</sup> Kwang-Mo Jung  
High Speed Network Research Center, Korea Electronics Technology Institute

### 요 약

최근에 인터넷의 보급이 확산되어지면서, 사용자들의 VOD, 홈 네트워킹과 같은 고품질의 서비스 요구도 증가하게 있는 추세이다. 특히 홈 네트워킹 시스템에서는 인터넷을 통해 대내의 가전제품과의 상호연결 및 제어까지 할 수 서비스의 개발에 박차를 가하고 있다. 이는 즉, 이기종 프로토콜을 수용하면서도 이들 간의 통신이 가능하게 하는 새로운 구조의 공통 프로토콜과 새로운 시스템 구조가 필요함을 의미한다. 본 논문에서는 이러한 공통 프로토콜과 새로운 칩의 구조 및 칩 내부의 패킷 스위칭을 위한 새로운 스위치 구조를 제안한다.

### 1. 서론

홈 게이트웨이 시스템에서 가장 중요한 점은 대내를 중심으로 LAN과 WAN의 상호 연결이라 할 수 있다. 현재 WAN 인터페이스로는 xDSL, 케이블모뎀 등이, LAN 인터페이스로는 wireless LAN, IEEE1394, LonTalk, Bluetooth, HomePNA, HomeRF 등의 프로토콜이 사용되어질 수 있다. 이 중에서 IEEE1394는 오디오 및 비디오 장치간의 고속 데이터 전송 및 제어를 위한 프로토콜이고[1], Bluetooth는 가전 기기간의 무선선을 이용한 저속 데이터 전송 프로토콜이다[2]. 또한 LonTalk는 가전 기기간의 제어만을 위한 프로토콜로서[3], 이들 각각의 프로토콜은 저속 및 고속 데이터 전송 또는 제어를 위한 각각의 특별한 응용분야가 존재한다. 따라서 홈 게이트웨이 시스템 설계에서 이들 간의 성공적인 통신을 위해서는 이들 프로토콜들의 차이점 및 특성을 파악하는 것이 중요하다. 본 논문에서는 이를 위한 홈 게이트웨이용 칩셋의 전반적인 구조와 함께 내부 스위치 구조에 대해서 제안한다.

### 2. 일반적인 홈 게이트웨이의 구조

ISO/IEC JTC1/SC25 WG1에서 제안한 홈 게이트웨이의 구조는 크게 3부분으로 나누어지며[4], 이들은 각각 Gateway Internal Protocol (GIP), WAN Gateway Interface (WGI), LAN Gateway Interface (LGI)이다. WGI, LGI는 각각의 프로세서와 메모리를 가지고 있어서 입력되는 패킷을 공통 프로토콜(Common Protocol, CP)로 바꾸어 주는 역할을 하고, GIP 내의 공통버스에서는 이들 공통 프로토콜 패킷을 목적지 노드의 GIP로 스위칭 해주며, 여기서 다시 공통 프로토콜 패킷을 해당 프로토콜로 바꾸어 주게 된다. 그러나 [4]에서는 자세한 H/W와 공통 프로토콜의 규격이 제시되어 있지는 않다.

특히 프로토콜 변환은 가장 큰 문제로, 변환시 지불해야할 시간적, 공간적인 노력은 하나의 프로토콜 규격 크기에 지수(exponential)배 이상으로 증가한다[5]. 따라서 이러한 홈 게이트웨이 시스템의 설계에 있어서 고려해야 할 점은 효율적인 공통 프로토콜의 설계와 이로의 효율적인 변환 알고리즘, 서로 다른 속도로 유입되는 트래픽의 조절, 보안 문제 및 이기종 프로토콜간 QoS를 보장해주는 스위치 구조의 설계가 가장 중요하다.

### 3. 제안된 칩셋의 구조

#### 3.1 칩셋의 구조

그림 1은 제안된 칩셋의 구조를 나타낸다. 여기서 입력된 패킷은 Anything-to-CP 변환기와 CP-to-Anything 변환기의 2개의 변환기를 지나게 되며, 각각의 변환기는 header 변환기와 payload 변환기로 구성되어 있다. header 변환기는 주소 테이블을 사용한 주소 변환 및 header 변환을 수행하고, payload 변환기는 RISC 프로세서 구조로 이루어져 있으며, ROM이나 PLA등의 비휘발성 메모리에 마이크로 코딩된 변환 코드가 수행되어 지면서 변환을 수행하게 된다. 이 payload 변환기는 간단하게는 layer 2-4 레벨에서 패킷을 처리하거나, 복잡한 동작의 경우 패킷 변환 같은 layer 5-7 레벨에서 패킷의 처리도 할 수 있어야한다 [6]. 이러한 프로토콜 변환 알고리즘은 현재 아직 개발 중에 있다. 각각의 인터페이스 블록은 칩 외부의 PHY/LINK 칩과의 연결 통로를 제공하여 주며, 외부로부터 입력되는 직렬 데이터를 byte 단위로 정렬해서 입력 버퍼에 보내거나 외부로 나가는 패킷에 대해서는 반대의 역할을 수행하게 된다. 트래픽 제어 블록은 트래픽 제어 알고리즘을 수행하여 버퍼 컨트롤러 하여금 입출력 버퍼내의 패킷을 통과 또는 폐기시키는 역할을 수행하게 된다. segmentation 블록은 입력되는 공통 프로토콜 패킷에 대해 256 byte의 여러 패킷으로 분할하여 스위치 블록으로 전송한다. 이 때 256 byte 패킷 내에는 공통 프로토콜 header를 포함하게 된다. 이와 같이 공통 프로토콜 header를 포함한 고정된 크기의 패킷을 스위치로 유입시키는 이유는 공유 메모리 방식을 사용하는 스위치에서 메모리의 효율적인 관리 및 QoS를 보장하는 스위칭을 하기 위해서이다. reassembly 블록은 스위칭된 분할 패킷에 대해 header 내의 sequence 번호에 따라 원래의 공통 프로토콜 패킷으로 합쳐주는 역할을 한다. 공통 버스는 각 노드의 segmentation 블록, reassembly 블록과 스위치 블록간에 분할 패킷의 공통 통로를 제공한다.

#### 3.2 공통 프로토콜 패킷

그림 2는 현재까지 개발된 공통 프로토콜의 규격을 나타내는 그림으로서 향후 칩 구조, 프로토콜 변환 알고리즘, QoS, 보안 알고리즘 등의 변화에 따라 수정될 수 있다. 본 논문에서는 공통 프로토콜에 초점을 맞춘 것이 아니므로 간략한 구조에 대해

서만 설명한다.

header에는 QoS, 보안, 트래픽 제어와 관련한 필드가 존재하고 최대 2,048 byte의 payload를 전달할 수 있도록 되어 있다. 이는 IP, LonTalk, Bluetooth 뿐만 아니라 IEEE1394의 400Mbps asynchronous 패킷 및 200Mbps isochronous 패킷을 하나의 공통 프로토콜에 전달할 수 있는 구조이다.

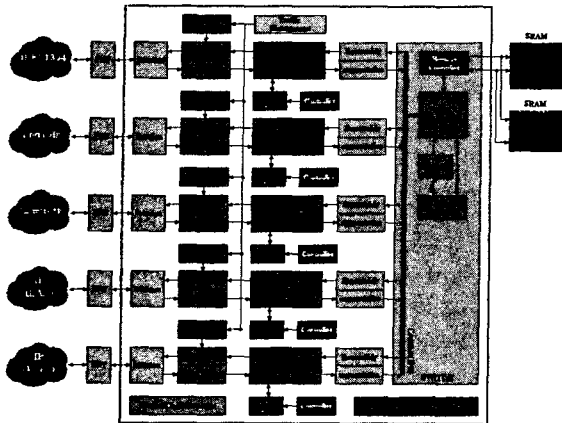


그림 1. 전반적인 칩 구조

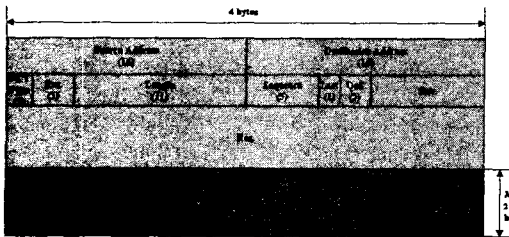


그림 2. 공통 프로토콜 구조

4. 스위치 구조

본 논문에서 제안하는 스위치 구조는 그림 3과 같이 칩 외부에 2개의 SRAM을 사용하는 구조로, 데이터 및 어드레스 버스와 콘트롤 신호를 공유하는 구조이다. 칩 내부의 메모리 컨트롤러에서 이들 메모리를 관리하게 된다. 하나의 SRAM은 패킷 메모리로 분할 공통 프로토콜 패킷의 저장 용도로 사용되어지며, 다른 하나는 패킷 메모리의 효율적인 관리를 위한 linked list 버퍼와, QoS 버퍼, priority 버퍼 용도로 사용된다. 전자의 경우 N개의 패킷을 저장하기 위해 256N byte 크기의 메모리가 사용되어지며, 후자의 경우 linked list 버퍼로 12N byte 및 QoS 버퍼와 priority 버퍼로 각각 36N byte 크기로 총 84N byte 크기이다.

4.1 패킷 메모리

실제 256 byte 크기의 분할 패킷 N개가 저장될 수 있는 곳이다. 하나의 패킷을 저장하거나 읽을때는 반드시 이에 해당하는 linked list가 갱신되어진다.

4.2 Linked List Buffer

하나의 분할 패킷의 읽기, 저장에 대해 12 byte의 linked list를 생성 및 삭제해주며, 이 list는 그림 4와 같이, 자신 이전의 list 주소 (PLA)와 자신 다음의 list의 주소 (NLA) 및 자기 자신이 몇 번째 list인가를 나타내는 값 (CLA)으로 구성되어

진다.

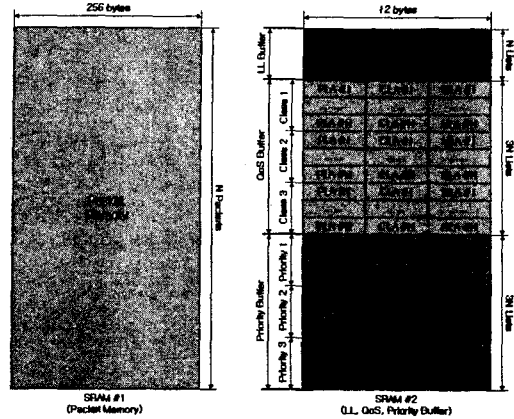


그림 3. 외부 SRAM 구조

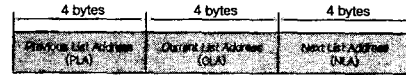


그림 4. linked list 구조

이들 list들은 Free-Space Linked Lists와 Used-Space Linked Lists의 두 분류로 나뉘어져 각각 linked list를 구성하고, 처음 칩의 구동 시에는 모든 linked list는 Free-Space Linked Lists로 연결되어 있게 된다.

해당 linked list가 가리키는 패킷 메모리의 주소는  $CLA \times 256$  (8-bit shift left)에 해당하고, list 자신의 address는  $CLA \times 12$  합으로써 얻어진다. 또한 패킷 메모리의 읽기, 쓰기에 따라 그림 5, 6과 같은 방법으로 linked list를 갱신한다.

4.3 QoS / Priority 버퍼

linked list 갱신 후 (그림 5), 입력 데이터 형태에 따라 class1, 2, 3로 나누어 해당 list를 QoS 버퍼에 저장하게 된다. 그리고 한번 더 입출력 라인 상태에 따라 priority를 적용하여 priority 버퍼에 저장한 후, 스케줄러의 스케줄링 알고리즘에 따라 priority 버퍼에서 list를 꺼낸 후, list에 해당하는 실제 분할 패킷을 읽어서 목적지 노드 쪽으로 보내주고, linked list를 다시 갱신하게 된다 (그림 6).

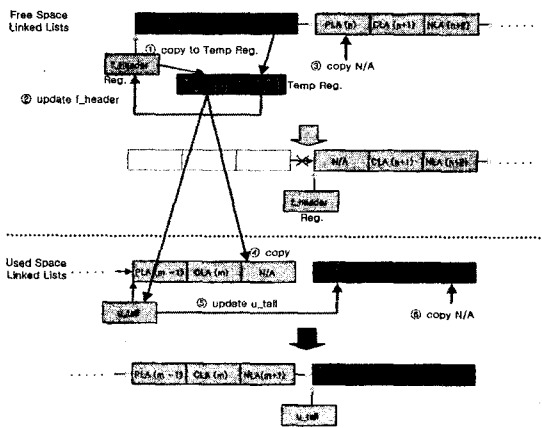


그림 5. 패킷 쓰기

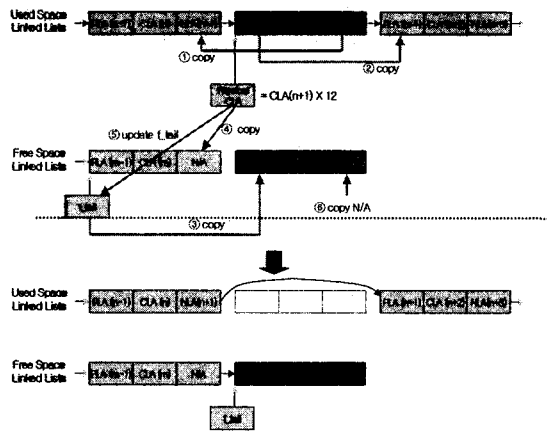


그림 6. 분할 패킷 읽기

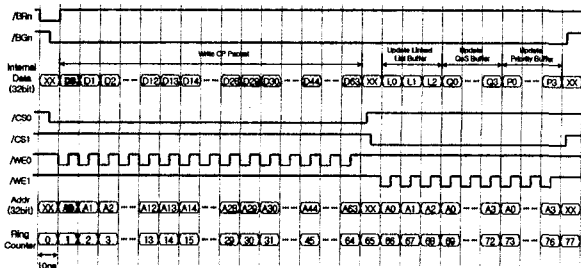
4.4 메모리 컨트롤러

그림 7과 같이 칩 내부에 메모리 및 공통 버스를 위한 신호를 발생하여 분할 패킷의 저장, linked list 갱신, QoS/Priority 버퍼 갱신을 하게끔 하는 역할을 한다.

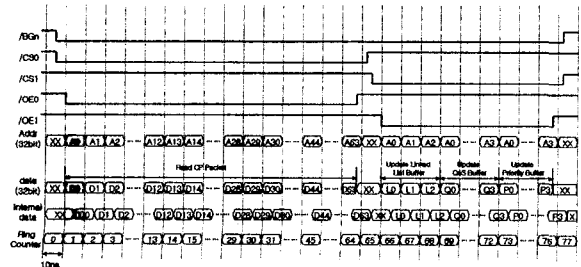
4.5 칩 성능 분석

표 1은 칩의 성능을 분석하기 위해 각 노드에서 최대 길이의 패킷이 최대의 속도로 들어오는 경우에 대해서 성능을 예측한 결과이다. 여기서 칩 내부를 인터페이스, Anything-to-CP (또는 CP-to-Anything), SAR (Segmentation & Reassembly), 스위치의 네 부분으로 나누어 계산한 결과이다. 패킷의 손실 없이 한 노드에서 다른 노드로의 입력부터 출력까지 패킷 변환 및 스위칭을 위해서는 아래의 수식을 만족해야만 하고, 표 1에서 보는 바와 같이 만족함을 알 수 있다.

$$P_1 \leq P_2 \leq P_3 \leq P_4$$



(1) 패킷 쓸 때의 신호



(2) 패킷 읽을 때의 신호

그림 7. 메모리 컨트롤러 신호

표 1. 칩 성능 분석

	인터페이스 (P <sub>1</sub> )	xx-to-CP (CP-to-xx) (P <sub>2</sub> )	SAR (P <sub>3</sub> )	스위치 (P <sub>4</sub> )
IEEE 1394	400 Mbps = 6,400 pkts/s	4MHz RISC CPU 512 cycles/pkt 7,812 pkts/s	1 pkt = 32 Seg. Pkts 4us/Seg.Pkt 7,812 pkts/s	Total # of incoming segmented packets = 382,788 Seg. Pkts/s
IP (LAN WAN)	100 Mbps = 8,738 pkts/s	5MHz RISC CPU 512 cycles/pkt 9,765 pkts/s	1 pkt = 6 Seg. Pkts 17us/Seg.Pkt 9,765 pkts/s	
Blue-tooth	2 Mbps = 155 pkts/s	1MHz RISC CPU 512 cycles/pkt 1,953 pkts/s	1 pkt = 7 Seg. Pkts 73us/Seg.Pkt 1,953 pkts/s	Switch Performance = 1.56 us/Seg. Pkt (write & read) = 641,025 Seg. Pkts/s
Lon-Talk	2 Mbps = 155 pkts/s	1MHz RISC CPU 512 cycles/pkt 1,953 pkts/s	1 pkt = 1 Seg. Pkt 512us/Seg.Pkt 1,953 pkts/s	

5. 결론

본 논문에서는 홈 게이트웨이를 위한 새로운 칩 구조 및 이를 위한 스위치 구조에 대해서 제안하였다. 이 칩은 WAN 인터페이스로 IP를, LAN 인터페이스로는 IP, IEEE1394, LonTalk, Bluetooth를 수용하고, 내부의 RISC 프로세서에 의해 패킷 변환 알고리즘이 수행되는 구조로 설계되었다. 이 패킷 변환 알고리즘을 위해 공통 프로토콜을 설계하였는데, 현재 계속 연구 중에 있으며, 변환 알고리즘 또한 연구 중에 있다. 스위치 구조는 공유 메모리 방식으로, 외부에 2개의 SRAM을 두어 하나는 패킷 저장용으로 다른 하나는 linked list와 QoS/Priority 버퍼 용도로 사용될 수 있도록 설계하였으며, 칩 내부 각 모듈별로 입력되는 패킷의 초당 처리 능력치를 계산하여 스위치 성능과 비교하였을 때 충분히 처리할 수 있는 것으로 분석되었다.

참고 문헌

- [1] D. Anderson, Firewire System Architecture, Addison Wesley, 1999.
- [2] K. Bray and C. Sturman, Bluetooth - Connect Without Cables, Prentice Hall, 2001.
- [3] Echelon Corp., "Lontalk Protocol Specification", Version 3.0.
- [4] ISO/IEC JTC1/SC25 WG1, CD1 15045-01, "Information technology - Interconnection of Information technology equipment - Architecture for HomeGate, the residential gateway(AHRG)", May 1999.
- [5] D. Lee, A. N. Netravali and K. K. Sabnani, "Protocol Pruning", in Proc. of the IEEE, Vol. 83, No. 10, October 1995.
- [6] L. Geppert, "The New Chips on the Block[Network Processors]", IEEE Spectrum, January 2001.
- [7] Dou, S. J. Jiang and K. C. Leu, "A Novel CAM/RAM Based Buffer Manager for Next Generation IP routers", in Proc. of the First IEEE International Workshop on Electronic Design, Test and Applications, 2002.
- [8] Noboru Endo, Takahiko Kozaki, Toshiya Ohuchi, Hiroshi Kuwahara and Shinobu Gohara, "Shared Buffer Memory Switch for an ATM Exchange", IEEE Transactions on Communications, Vol. 41, No. 1, January 1993