10GE 스위치 시스템에서의 이중화 IPC 설계 및 구현

조규인*, 김유진**, 김준식*, 조경록***
*이스텔시스템즈 가입자망 개발1팀 **한국전자통신연구원 10기가 H/W팀
***충북대학교 전기전자공학부

Architecture of the Duplicate IPC Network for the 10 Gigabit Ethernet Edge Switch System

Gyuin Cho*, Youjin Kim**, Junsik Kim*, Kyoungrok Cho***
Access Network R&D Team, Eastelsystems*
10GE H/W Team, Electronics and Telecommunication Research Institute**
Dept. of Electrical & Electronics Engineering, Chungbuk National University***
E-mail : jki75760@etri.re.kr*

Abstract

본 논문은 이스텔시스템즈와 한국전자통신연구원(ETRI)이 공동개발한 QoS 기반의 10기가비트 이더넷 포트를 가지는 에지 스위치 시스템을 개발 하면서 적용된 분산된 스케일러블 이중화 IPC구조를 제안한다. 제안된 분산된 스케일러블 이중화 IPC구조는 라우팅 관련 로드를 분산하고 데이터 전달 능력을 향상시켜 라우팅의 성능을 개선시키는 것이다. 이 기능이 가능한 것은 라우팅 테이블 생성 및 관리의 분산을 위하여 전이중 방식의 고속이더넷 스위치를 이용한 이중화 IPC구조로 설계되었기 때문이다. 본 논문에서는 분산된 스케일러블 이중화 IPC 구조에 대한 내용을 설명하고 그에대한 구현 방법을 설명한다. 제안된 분산된 스케일러블 이중화 IPC는, 차후 공동개발되는 160Gbps급의 10기가비트 이더넷 백본 스위치 시스템에도 적용함으로써 보다 신뢰성 있는 시스템의 설계 효과를 가져올 것으로 생각된다.

1. 서론

기존의 일반적인 마이크로 프로세서를 이용한 네트워크 장비의 성능 한계를 벗어나기 위해, 현재 고속의 패킷 스위칭 및 라우팅을 구현할 수 있고, Layer3 스위칭 및 Diffserv.를 하드웨어로 지원가능한 네트워크 전용 프로세서인 NP(Network Processor)를 사용한 네트워크장비의 개발이 보편화 되어지고 있다[1].

NP를 이용한 시스템의 구조는 일반적으로 패킷처리 기능을 담당하는 NP와 제어 및 관리 기능을 담당하기 위해 사용되는 일반적인 마이크로프로세서인 호스트 프로세서(CP: Control Processor)의 구조가 일반적이다. 이런 구조에서 사용되는 인터페이스로써, PCI 버스등이 제공 된다[2]. 패킷처리를 위한 NP가 헤더파싱, 패턴 매칭, 비트필트 조작, 테이블 룩업, 패킷 수정 및 트래픽 관리 등을 하드 와이어드 속도(hard-wired speed)를 지원 하는 기능을 하는데 반하여, CP는 시스템 초기화를 수행하고, 포워딩 테이블에 엔트리를 추가하고, 멀티캐스트 그룹들을 관리하고, 버퍼관리에서 임계값 등을 바꾸는 관리 및 제어 기능을 수행을 한다. 근래에는 NP의 부가 기능으로 IP포워딩, 프로토콜 변환 기능, QoS, 보안, 트래픽 대역폭 할당 기능, VoIP 기능들이 추가되고 있는데, 이를 위해서는 상위 레이어(L4~L7)에 대한 패킷 프로세싱을 요구한다. 이를 위하여 CP가 외부 상위 프로토콜 스택(L4~L7)을 지원하는 기능도 한다.

NP를 사용한 근래의 일반적인 라우터의 경우, 라우팅 관련 프로토콜 처리 및 라우팅 테이블 유지등을 전담하는 라우팅 프로세서, 입출력 패킷을 스위칭 해주는 스위치패브릭, 네트워크 인터페이스 및 포워딩 기능을 처리하는 라인카드 모듈로 구성이 된다[3].

이러한 근래의 일반적인 라우터 구조에서, 라우팅 기능을 1개의 CP내에 구현하여 PCI버스를 통한 n개의 NP를 제어하는 중앙집중식 시스템 구조와 각 라인카드별로 CP를 두어 라인카드에 있는 NP를 이 CP가 제어를 하고 n개의 CP를 다시 메인프로세서(MP)가 IPC기능을 통하여 관리를 하는 다중 분산 시스템의 경우를 생각 해 볼 수 있다. 다수의 라인카드 모듈로

구성되는 대용량 다중 분산 시스템에서는 타 모듈의 프로세서와 필요한 정보를 송수신하기 위한 프로세서간 통신(IPC: Inter Processor Communication)이 필요하다[4].

본 논문에서는 리눅스 기반의 64Gbps급의 10기가비트 이더넷 에지 스위치 시스템을 개발하면서, NP와 CP간의 효율적인 IPC구조에 대하여 기술하고, 차후 160Gbps급의 10기가비트 이더넷 백본 스위치 시스템으로 안정적인 확장이 가능한 동작을 위하여 스케일러블한 이중화 IPC구조에 대하여 제안한다.

2장에서는 라우팅 기능을 1개의 CP내에 구현하여 PCI버스를 통한 n개의 NP를 제어하고 IPC메시지 통신을 하는 중앙 집중식 IPC구조와 이를 위한 소프트웨어 구조를 설명한다. 3장에서는 각 라인카드 별로 CP를 두고 NP를 제어를 하고 n개의 CP를 100Mbps 고속이더넷 스위치를 통해 제어하는 분산된 스케일러블 이중화 IPC 구조와 이를 위한 소프트웨어 구조를 설명한다. 4장에서는 실제 설계 및 구현된 결과물을, 그리고 마지막 5장에서는 결론 및 향후 추진 방향을 기술한다.

2. PCI 버스를 이용한 중앙 집중식 IPC구조

초기 라우터의 구조는 메인프로세서가 공유된 버스를 이용하여 입,출력되는 패킷에 대한 라우팅 계산 및 포워딩의 모든 태스크를 중앙 집중적으로 수행 하였다. 이러한 초기 라우터는 메인프로세서의 집중된 패킷 전송의 체증과 과중한 부하 발생을 일으켜 비효율적인 문제를 야기 시켰다. 그 이후로는 그림 1. 에서와 같이 공유버스 대신에 스위치 패브릭을 이용하게 되었고, 고속의 패킷 포워딩은 라인카드의 NP가 담당하고 라우팅 계산 및 테이블 갱신은 CP가 담당을 하게 되어 분산형 구조의 라우터의 모습을 가지게 되었다.

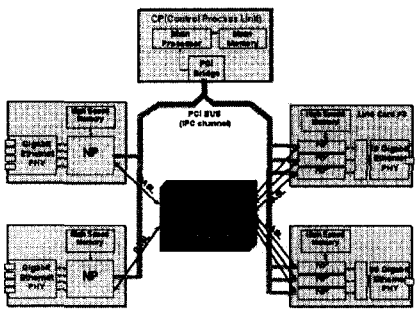

그림 1. 중앙 집중식 단일 IPC구조

NP에서 제공하는 PCI버스를 통하여 CP와 통신을 하는데, PCI버스는 33MHz, 32Bits을 기준으로 132Mbytes/sec(=1.056Gbps)의 전송 속도를 가진다. 이 경우 49개의 PCI버스 신호선들이 요구가 되어 진다. 고속의 PCI버스를 이용하여 모든 NP를 제어하는 CP는 PCI디바이스 갯수에 따라 추가적으로 PCI브리지 칩의 사용이 요구가 된다. 이러한 버스의 제한 사항은 중앙 집중화된 라우팅 테이블의 관리를 할 수 있다는 큰 장점에도 불구하고, 구현 및 설계의 복잡성을 가지게 된다. 이 경우 개인 및 소기업용 라우터급인 엑세스 라우터의 구조에는 적합할 수가 있으나, 에지 및 백본 라우터에서는 확장성 및 시스템 이중화적인 측면에서 효율성이 떨어 질 수 있다. 그림 1.에서 하드와이어드 패킷은 DASL[5]이라는 NP와 스위치 패브릭간의 인터페이스를 경유하여 이루어 지고, 관리 및 제어 처리는 PCI버스를 통하여 이루어진다. 이 경우 패킷을 쓰레드(thread)형태로 처리하는 NP의 관리테스크와 CP간의 IPC의 메시지 교환은 PCI버스를 경유하여 이루어 진다[6].

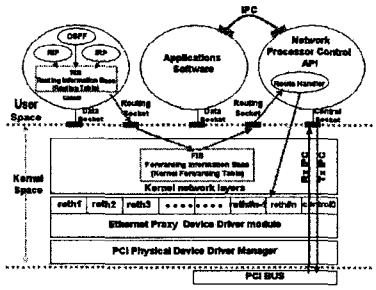

그림 2. 중앙 집중식 소프트웨어 구조

그림 2.는 상기 구조에서 구현된 소프트웨어 구조 이다. 리눅스 기반의 시스템에서 라우팅 프로토콜로 사용될 수 있는 프로토콜중 GateD는 커널에서 포워딩 테이블을 갱신 하는데 사용된다. GateD는 몇 개의 내부 라우팅 프로토콜[Internal Routing Protocols (IRP)]로 구성이 된다. OSPF(Open Shortest Path First)와 같은 IRP들은 IRP가 동작하는 영역에서 라우팅을 학습하고 그 결과를 중앙 라우팅 테이블에 놓는다. 일반적으로 GateD의 중앙 라우팅 테이블을 RIB (Routing Information Base)라 한다. GateD는 RIB를 검사하고 가장 최상의 라우팅 경로를 선택한다. 이러한 라우팅 경로는 시스템 커널 포워딩 테이블에 놓이는데, 이러한 포워딩 테이블을 일반적으로 FIB(Forwarding Information Base)이라 부른다. 이러한 테이블은 일반적으로 데이터 패킷을 포워딩 하는데 사용 된다.

데이터 패킷이 포워딩 결정을 위해 CP의 커널을 이용하는 것을 막기 위해서, FIB는 NP의 L3 tables

메모리에 복사되어 포트로부터 포트로 포워딩 되도록 한다. 유사한 방법으로 NP의 메모리에 L2 의 ARP테이블을 업데이트 하도록 한다. 모든 포트는 프록시 디바이스 드라이버를 통하여 커널에 통신 포트(reth1~n)으로 인식이 된다. 이 프록시 디바이스 드라이버는 라인카드의 모든 포트목록을 나타내도록 하는 드라이버로, PCI버스를 하나의 네트워크 포트로 인식하여 NP와 CP간의 IPC통신을 담당하게 하는 제어용 통신 포트(control0)이다.

NP Control API가 하는 역할은 NP가 사용하는 실시간 정보인 L3의 포워딩 엔트리와 ARP를 다이나믹 하게 업데이트 하는데, 이러한 일들은 독립적인 프로세스로 작업을 분리하면, 상당한 양의 오버헤드가 발생하기 때문에 일률적으로 NP Control API 자체에서만 수행한다[6].

라우트 핸들러는 리눅스 커널 라우팅 테이블과 NP의 포워딩 테이블 사이의 인터페이스 역할을 담당하는 모듈이다. 커널 라우팅 테이블은 명령어 입력이나 GateD 등을 통해서 스태틱 라우트에 의해 변경된다. NP Control API는 이러한 변경사항을 라우팅 소캣을 통해 메시지를 주고 받는다.

3. 분산된 스케일러블 이중화 IPC구조

분산형 라우터 시스템 구조에서 가장 중요한 동작 중의 하나가 CP에서의 라우트 최단 경로의 계산과 함께 라우팅 테이블의 관리이며, NP에서의 패킷 분류 및 스케줄링, 효율적인 목적지 검색, QoS기능 지원의 포워딩 테이블 갱신이다.

이러한 동작을 신뢰성 있게 하기 위해서는 라우팅 기능과 포워딩 기능의 확실한 분리가 매우 중요하다. 그러나, 네트워크의 불안정에 의한 라우트 플랩(route flap)발생시 NP와 CP간의 잦은 IPC 트래픽 증가등의 문제와 장애요인 발생시의 신속한 해결을 위해서는 라우팅 프로세싱의 이중화와 함께 패킷경로의 이중화, IPC경로의 이중화등을 고려 해야 한다.

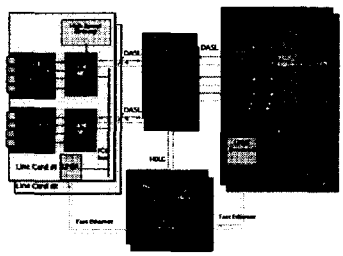

그림 3. 스케일러블 이중화 IPC구조를 가지는 10기가비트 이더넷 에지 스위치 시스템 구조

이중화의 고려는 장비개발의 가격 상승 요인과 더불어 고장요소의 증가를 가져 올 수 있기 때문에, 확장 가능한 구조의 설계가 중요시 된다. 이러한 문제는 이중화구조가 단일 플랫폼에서 사용자의 서비스 제공 용량에 따라 용이하게 용량을 변경 할수 있도록 하여, 에지급부터 백본 코어급까지의 확장이 가능하게 하는 스케일러블(scalable) 라우터의 개발이 요구 된다[3].

본 논문에서는 스케일러블 라우터 구조에 적용가능한 이중화 IPC 구조를 제안한다. 그림 3.과 같이 10기가비트 포트를 가지는 라인카드와 1기가비트 포트를 가지는 라인카드에 각각 CP를 두고 이를 두개의 IPC모듈을 통하여 한 개의 MP가 이를 관리 하도록 한다. 각각의 IPC모듈은 100Mbps 전이중 방식의 이더넷 스위칭 역할을 하는 모듈이다. IPC의 이중화는 CP와 MP를 포트베이스의 VLAN으로 구성하는 논리적인 이중화 뿐만 아니라, 물리적인 경로 이중화를 지원 하도록 하였다. 또한, 트래픽 경로의 이중화를 위하여 스위치 패브릭간의 이중화도 고려 하였다. 이러한 이중화의 고려는 MP의 라우팅 테이블과 CP에서 관리되는 NP의 포워딩 테이블의 신뢰성 있는 관리를 제공하게 된다[7].

두개의 MP는 Active MP#1와 Standby MP#2로 구성하여 라우팅 테이블의 복사를 주기적으로 MP#2에 할수 있는 IPC경로(Duplicate IPC)를 별도로 구성하였다. MP#1의 장애 요인 발생시 MP#2로의 사용이 가능하고 이때 복사된 테이블을 다시 이용할 수 있어, 테이블 손실을 최소화 시킬 수 있다.

10기가비트 이더넷 에지 스위치 시스템에서 일반적으로 10기가비트 이더넷 포트는 백본 스위치 시스템과 연결하는 업링크(Up-link)의 포트로 사용될 가능성이 크다. 따라서, 스케일러블 라우터의 중요한 제어요소인 테이블 동기 기능을 위하여 백본 스위치가 에지 스위치 노드들이 하나의 스위치 시스템으로 보이도록 하여야 한다. 이를 위하여 각 에지와 백본간의 광역적인 라우팅 테이블 교환이 이루어져야 한다. 따라서 제안된 IPC구조는 부가적인 기능으로 이를 위해 각 IPC모듈에 각 에지 시스템의 MP를 서로 연결 할 수 있는 IPC포트(Global IPC)를 가지도록 하였다.

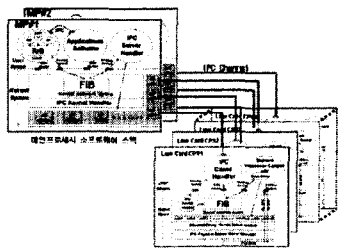

그림 4. 이중화 IPC구조의 소프트웨어 구조

그림 4.는 제안된 소프트웨어 구조를 나타낸다. 라인카드의 각각의 CP를 이중화 IPC 채널을 통하여 MP의 프로토콜 스택과 연결된다. MP에서는 수행된 라우팅 테이블 정보는 MP의 FIB를 거쳐 IPC 서버 핸들러를 통해 각각의 CP와 IPC로 연결되어 전송 된다. 전송된 IPC 메시지는 CP의 IPC클라이언트 핸들러를 통해 리눅스 커널에 있는 FIB로 전해지며, NP Control API를 통하여 NP가 포워딩 테이블로 사용하는 고속메모리에 위치하게 된다. 이러한 방법은 이중화 IPC구조를 제외하면 그림 2.에 설명된 소프트웨어 스택의 동작과 유사하다.

경우에 따라 IPC메시지가 일반적인 이더넷 프레임인 64bytes 에서 1518bytes를 넘을때는 Extra long 프레임 기능(1530bytes까지 사용 가능)을 설정하여 사용한다. 만약 1530bytes보다 긴 경우의 프레임은 IPC모듈에 있는 MAC의 DA 첫번째 바이트부터 FCS의 마지막 바이트까지의 프레임의 길이를 판독하여 처리를 한다.

그림 5.는 제안된 스케일러블 이중화 IPC를 통한 Throughput 결과이다.

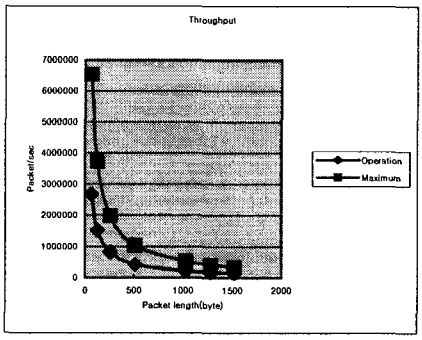

그림 5. 제안된 스케일러블 이중화 IPC를 통한 Throughput 결과

위 그림을 통해 제안된 스케일러블 이중화 IPC는 프레임의 손실없이 신뢰성 있는 IPC통신을 통해MP의 기능을 분산시킴으로써 보다 효과적인 시스템의 구성이 가능함을 보여준다. 이것은 기존의 라우터가 PCI버스를 통한 라우팅 기능과 포워딩 기능을 모두 처리하기 때문에 발생되었던 트래픽 집중 문제로인한 네트워크의 성능 저하를 가져왔던 문제를 MP와 CP로의 RT와 FT를 분리함으로써, CP의 FT를 통한 하드웨어 기반의 포워딩 기능을 가능케 함으로써 wire-speed로 라우팅 기능을 수행할 수 있음을 보여준다. 여기서, MP는 여러가지 라우팅 프로토콜(RIP, BGP, OSPF)을 수행하여 RT를 만들어내고, 만든 RT를 각각의 CP에 맞는 최적화 된 FT를 스케일러블 이중화 IPC를 통해 전달됨을 보여준다.

4. H/W 설계 및 구현 결과물

그림 6.은 분산된 스케일러블 이중화 IPC의 실제 설계 및 구현된 사진을 보여준다.

5. 결론 및 향후 추진 방향

인터넷 이용자의 급속한 증가에 따른 인터넷의 데이터 증가는 일반적으로 라우터에서 데이터 전달의 병목 현상을 일으켜 망의 성능에 큰 영향을 미치고 있다.

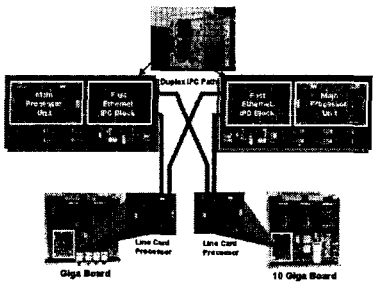

그림 6. 실제 구현된 분산된 스케일러블 이중화 IPC

본 논문에서는 이런 문제점을 해결하기 위해 QoS 기반의 64Gbps급의 10기가비트 이더넷 에지 스위치 시스템을 개발 하면서 적용된, MP와 CP간 분산된 데이터 전달 기능을 하는 스케일러블 이중화 IPC를 통해 시스템의 성능을 향상시키기 위한 방법을 제안한다. 이는 MP에서 RT를 관리하고 IPC를 통한 CP에서의 FT를 관리하는 기능을 설명하였다. 이로 인하여 라우터의 일이 분산되어 망의 성능을 개선시킬 수 있을 것으로 보인다. 이는 차후에 개발되는 160Gbps급의 10기가비트 이더넷 백본 스위치 시스템에도 적용하여 보다 안정적인 시스템의 동작을 확보할수 있는 계기를마련하였다.

추후에는 상기 구조에서 리눅스 기반의 효율적인 이중화 IPC 메시지 구성과 동작 절차의 개발이 지속적으로 연구 되어야 할 것이다.

참고문헌

[1] 김봉완, 이형호 "네트워크 프로세서의 응용과 표준화 동향" 電子工學會誌 第28卷 第10號,p94~p101, 2001年10月

[2] Linley Gwennap, Bob Wheeler "A Guide to Network Processors", MicroDesign Resources, 1st Edition,2000.

[3] 이형호, 김봉완, 안병준 "테라비트 라우터 기술", Telecommunications Review, 第11卷 第2號, p237~p247, 2001年4月

[4] Bup Joong Kim, "Design and Implementation of IPC Network in ATM Switching system", IEICE TRANS. COMMUN. VOL.E83-B

[5] IBM Packet Routing Switch PRS28G DataSheet,Ver1.7 (www.ibm.com)

[6] IBM NP4GS3 Advanced Software Offering Control Application Programming Interfaces Reference November 15, 2001 (www.ibm.com)