라우터의 성능향상을 위한 제어 데이터 전송방법 개선

윤천균*
*호남대학교 정보통신대학
e-mail : chqyoun@honam.ac.kr

An Improvement on Control Data Transmission Method for Performance Elevation of Router

Chunkyun Youn*
*Dept. of Internet software, Honam University

요 약

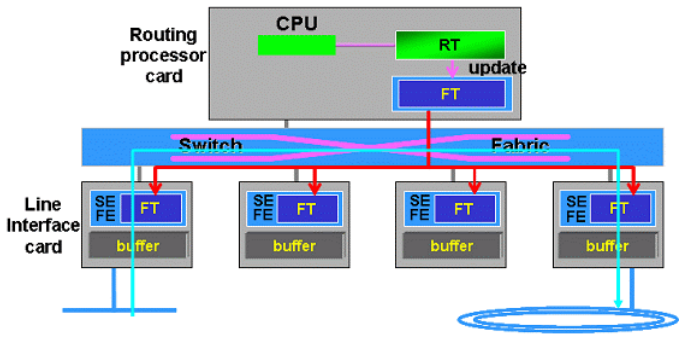
최근의 대용량 다중 분산 라우터 시스템은 다수의 라인 인터페이스 모듈들과 라우팅 처리 모듈, 스위칭 패브릭 모듈로 구성되어 있고, 고속의 패킷 스위칭 및 라우팅을 구현하기 위하여 일반적으로 입력 패킷을 외부로 전송하기 위한 기능과 제어 및 관리 기능을 담당하는 기능으로 분리하여 실행되고 있다. 이러한 라우터에서는 내부 모듈들의 프로세서들 사이에 정보 송수신을 위해 프로세서 간 통신(IPC : Interprocess Communication)이 이용되고 있다.

라우터의 기능 중 제어 및 관리 기능은 신속한 처리를 위하여 UDP/IP 방식의 IPC 가 사용되고 있는데, 이 UDP/IP 방식을 개선 방안을 제안하고 prototype 시스템을 구현하여 시험한 결과 라우터의 데이터 round trip 시간과 throughput 이 각각 15.1%, 4.3%의 개선되어 라우터의 성능이 향상되었다.

1. 서론

일반적으로 대용량 라우터의 경우 그림 1 과 같이 다수의 라인 인터페이스 카드와 라우팅 프로토콜이 수행되는 라우팅 프로세서 카드, 그리고 이들간을 연결해 주는 스위치 모듈로 구성된다[1,2]. 이와 함께 시스템의 신뢰성을 높이기 위해서 프로세서 카드와 스위치 카드 같은 주요 컴포넌트들은 이중화되어, 동작하던 카드에 이상이 발생하더라도 대기하던 백업용 카드가 이후의 처리를 이어받아 중단 없는 운용이 이루어지도록 하고 있다[3].

대용량 라우터에서는 패킷 forwarding 을 고속으로 분산 처리하기 위해서, 라우팅 프로토콜에 의해 생성된 RT(Routing Table) 정보가 각 라인카드에 FT (Forwarding Table) 형태로 복사되며, 라인카드로 들어오는 패킷들은 이 FT 를 룩업하여 전달되어야 할 라인카드와 인터페이스를 결정한 후 데이터를 전송한다[2,4,5].



(그림 1) 대용량 다중 분산형 라우터의 하드웨어 구조

라우터에서의 패킷은 자신을 시발점이나 종착점으로 하는 컨트롤 패킷과 이웃한 라우터로 단순히 forwarding 되는 데이터 패킷으로 구분되는데, 이들 패킷이 전달되는 경로를 각기 제어경로와 데이터경로라고 하며, 데이터경로를 지나는 패킷 중에서 라인카드의 firmware 에서 처리되지 못하는 예외적인 데이터

패킷에 대하여는 프로세서 카드까지 연결된 슬로우 패스를 거쳐서 처리된다[5,6].

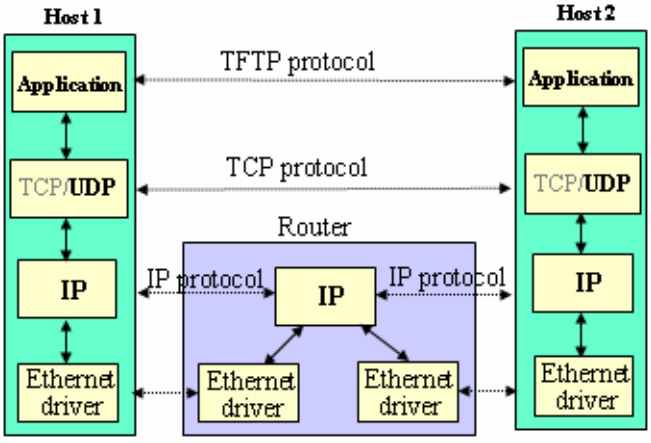
대용량의 고속 라우터는 기능상으로 시스템 초기화, forwarding 테이블에 엔트리 추가, 멀티캐스트 그룹들의 관리, 버퍼의 임계값을 갱신하는 버퍼관리 등의 관리 및 제어 기능과 데이터를 고속으로 전달하기 위해 헤더 파싱, 패턴 매칭, 비트필드 조작, 테이블 탐색, 패킷 수정 및 트래픽 관리, IP Forwarding, 프로토콜 변환, QoS, 보안, 트래픽 대역폭 할당, VoIP 기능 등을 처리하는 데이터 forwarding 기능으로 구성된다. 이러한 기능 수행을 위하여 다중 분산형의 라우터 시스템에서는 모듈들의 프로세서들 간의 정보 송수신을 위해 IPC 를 이용한다[7,8].

일반적으로 IPC 는 셀 버스(Cell Bus), HDLC(High Level Data Link), ATM(Asynchronous Transfer Mode) 스위치 버스, 이더넷 스위치 버스를 이용한 방식을 사용한다. 이중 셀 버스와 HDLC 방식은 공유버스 방식으로 일대일 통신 방법을 사용하는 ATM 스위치 버스와 이더넷 스위치 버스에 비해 처리성능이 떨어지는 단점이 있다. 성능이 비교적 우수한 ATM 스위치 버스와 이더넷 스위치 버스 중에서 저가이면서 사용이 간편한 이더넷 스위치를 이용한 IPC 채널이 주로 사용된다[7,9]. 본 논문에서는 이러한 기술적인 요구사항을 고려하여 다중 분산 시스템에서의 효율적인 이더넷 IPC 메커니즘을 제안하고자 한다.

2. 효율적인 이더넷 IPC 메커니즘의 제안

2.1 기존 IPC 방식의 문제점

라우터는 필요한 정보들을 신뢰성 있는 전송을 위한 TCP/IP 프로토콜과 신속한 처리를 위한 UDP/IP 프로토콜을 병행하여 사용한다. 이러한 정보들 중 관리 및 제어 기능을 위한 OAM(Operation, Administration, Maintenance) 정보와 라우팅 테이블 등의 정보는 신속한 처리를 위하여 UDP/IP 를 사용하는 TFTP(Trivial FTP)를 이용하여 전송한다. 따라서 이 종류의 데이터 전송은 그림 2 와 같은 TCP/IP 프로토콜 계층 중에서 UDP/IP Encapsulation 과정을 거쳐 송수신된다[7,9].



(그림 2) UDP/IP 를 이용한 관리/제어 정보 처리도

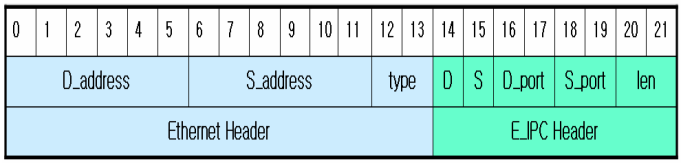
물론 TCP/IP 를 사용하는 것보다 속도 측면에서 유리하지만, 이 역시 여러 계층에서의 복잡한 처리 과정을 거쳐야 하기 때문에 폭주하는 인터넷 트래픽을 처리하는데 상당한 처리 시간이 소요되어 병목현상의 원인이 되기도 한다.

2.2 개선된 E-IPC 방식 제안

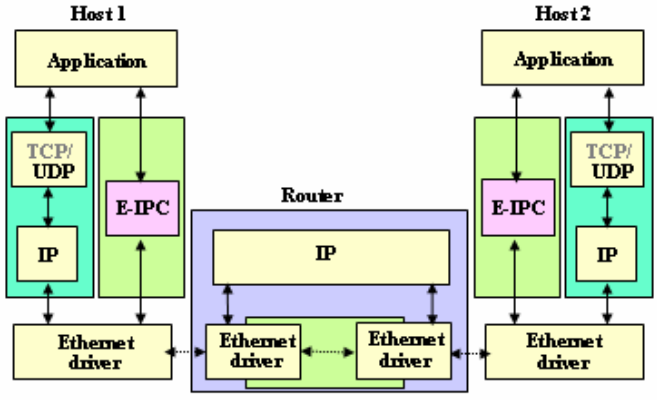
본 연구에서 제안하는 E-IPC 통신 메커니즘은 한 AS (Autonomous System) 내에서 라우터에 직접 연결된 네트워크에 대해서는 해당 라우터가 송수신지의 IP 주소, MAC 주소, 슬롯번호, 포트번호 등 각 인터페이스에 대한 상세한 정보를 보유하고 있다.

본 논문에서 개선의 대상으로 하고 있는 라우터의 관리 및 제어기능 목적의 메시지 전송은 한 AS 내에서 라우터 간 또는 시스템과 라우터 간의 통신에 해당되므로, 해당 라우터가 알고 있는 정보를 이용하면 IP 와 MAC 주소 모두를 사용하지 않고 그림 3 과 같이 MAC 주소, 슬롯번호, 포트번호 등을 이용해서 전송이 가능하다. 이를 위해 E_IPC 에서는 그림 3 과 같이 개선된 이더넷 프레임을 만들 수 있도록 하고, 라우터의 각 프로세스들은 이 프레임을 이용하여 라인 인터페이스 카드, 라우팅 프로세서 카드와 스위치 모듈 간에 그림 4 와 같은 E_IPC 계층을 통하여 데이터를 forwarding 한다. E-IPC 계층 이용 시 이더넷 프레임에 추가되는 E-IPC header 의 정보는 다음과 같다.

- D : 목적지 슬롯(destination slot) 번호
- S : 소스 슬롯(source slot) 번호
- D_port(destination port) : 목적지 응용프로그램 포트 번호로 메시지를 수신하는 응용프로그램
- S_port(source port) : 소스 응용프로그램 포트 번호로 메시지를 송신하는 응용프로그램
- len : 프레임의 전체 크기



(그림 3) 개선된 이더넷 프레임 format



(그림 4) E-IPC 를 이용한 관리/제어 정보 처리도

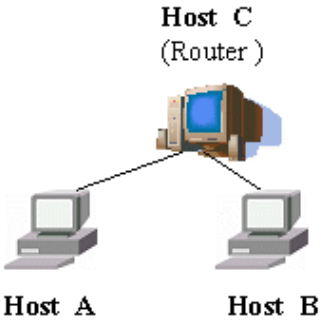
제안된 개선안에서 메시지의 처리과정은 관리 및 제어와 관련되어 신속한 처리가 요구되는 메시지는 그림 4 에서와 같이 E-IPC 프로토콜 계층을 통해서 처리할 수 있도록 하고, 외부로부터 들어온 데이터 패킷을 처리하는 forwarding 기능용 메시지는 기존의 방식대로 UDP/IP 계층을 통해서 처리한다. 위와 같이 메시지의 종류에 따라 분리하여 전송하는 방법을 통해 2.1 절에서 언급한 기존 통신방식의 문제점인 여러 계층에서의 복잡한 처리 때문에 발생되는 처리시간 지연을 개선할 수 있다.

제안된 개선안을 구현하기 위해 여러 라우터 내의 프로세스들이 다른 프로세스와 통신할 수 있도록 하는 IPC 를 이용한다. 본 연구에서는 리눅스 시스템에서 제공되는 여러 가지 IPC 구현 방법 중에서 동기화 문제와 블로킹 문제를 해결할 수 있으며, 소규모에서 대규모까지 데이터 처리가 용이하고, 프로그램 코드가 간단한 메시지 큐를 이용하였다.

3. 성능 비교시험 및 결과분석

3.1 시험 환경 및 조건

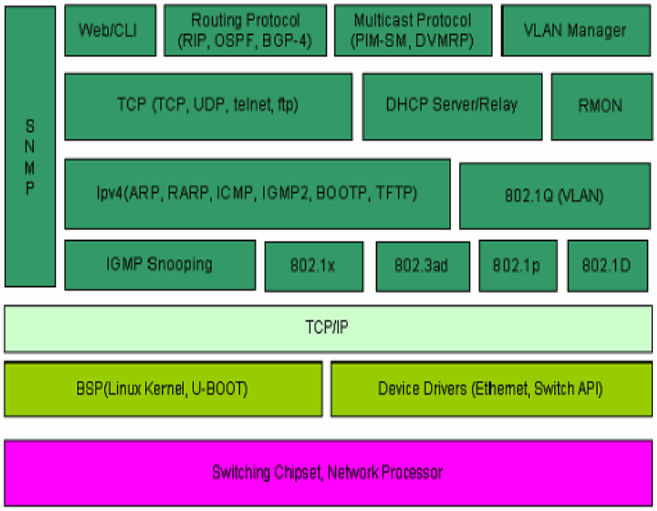
기존의 라우터에서 사용 중인 UDP/IP 통신 방식과 본 연구에서 제안한 E-IPC 방식의 성능비교를 위하여 그림 5 와 같은 prototype 시스템을 구축하였다. 시험 시스템은 외부와의 네트워크를 단절하고 2 대의 시스템 A 와 B 를 2 개의 이더넷 인터페이스를 가진 라우터 기능을 구축한 시스템 C 에 직접 연결하여 시험함으로써 외부로부터의 불필요한 트래픽을 차단하여 성능비교 시험 시 오차 발생 가능성을 최소화 했다.



(그림 5) Prototype 시스템의 하드웨어 구성도

정확한 성능비교를 위해 두 가지 통신 방식이 동일 조건에서 시험을 해야 하므로 모든 시스템에 그림 6 과 같은 소프트웨어 구성이 되도록 하였다. 이들 중 라우터와 관련된 소프트웨어는 맨 하위 계층인 Switching Chipset 와 Network Processor, 2 계층인 BSP(Bootstrap Processor)에서 U-BOOT 와 Linux Kernel, Device Driver, 3 계층의 TCP/IP 와 맨 상위 계층인 Application 부분으로 나누어진다.

본 시험에 사용된 각 시스템의 운영체제는 Red Hat Linux 7.3 을 사용하였다. 각 시스템의 하드웨어 사양은 Pentium IV-1.7G, 512 DDR RAM 및 Seagate Barracuda ATA IV 5400rpm, 40G HDD, 100Mbps 이더넷 카드를 사용하였다. 시스템 C 는 라우터 기능만을 할 수 있도록 하였으며, 다른 두 시스템 A 와 B 는 TCP/IP 의 모든 기능과 라우터 기능을 수행할 수 있도록 하였다. 시스템 A 에는 성능 비교 시험을 위해 부하를 생성함과 동시에 모니터링이 가능한 OpenSTA 1.4 를 설치하였다 [10].



(그림 6) Prototype 시스템의 소프트웨어 구성도

두 가지 통신방법의 Round trip 시간 비교 시험을 위해 표 1 과 같이 64, 256, 512, 1024, 1536 bytes 의 패킷을 시스템 A 에서 1,000 개씩 전송하여 시스템 C (라우터)를 경유하여 시스템 B 에 도착한 후 다시 역으로 되돌아와 시스템 A 에 도착하는데 소요되는 시간을 측정하였다. UDP/IP 와 E_IPC 방식에서의 전송속도(Throughput) 비교 시험을 위하여 호스트 A 에서 B 호스트로 64, 256, 512, 1024, 1536 bytes 의 패킷별로 100Mbps 데이터 량을 전송하여 호스트 B 에서 두 가지 방식의 전송속도를 측정 비교하였다.

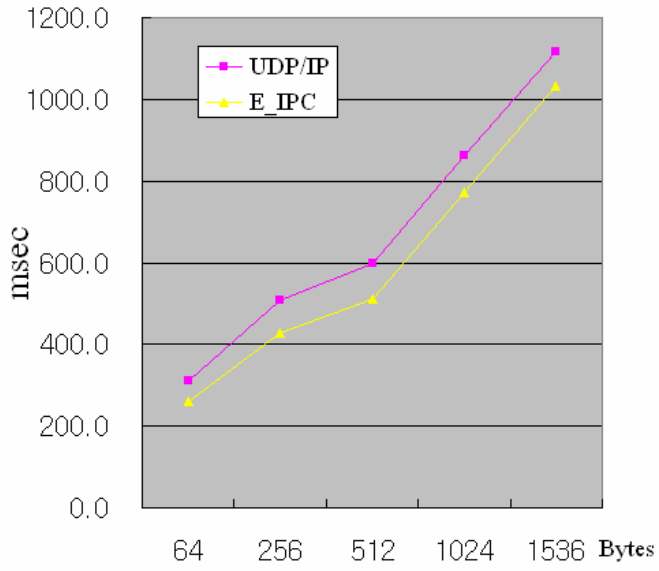
3.2 시험 결과 및 분석

1) Round trip 시간

표 1 과 그림 7 은 UDP/IP 와 E_IPC 의 RTT 성능비교 시험의 측정 결과이다. 제안한 E_IPC 의 프로토콜이 UDP/IP 에 비해 시험에 사용한 모든 크기의 패킷에서 RTT 평균 약 15.1% 빨라졌으며, 패킷 크기가 1536 바이트일 때 82.0 msec 단축되었다. 결과적으로 제안한 E_IPC 를 이용하여 처리할 경우 라우터의 성능이 향상되었음을 알 수 있다.

<표 1> Round trip 시간 시험 결과

Sended Byte		64	256	512	1024	1536
aver_RTT(msec)	UDP/IP	311.3	507.3	599.4	862.9	1115.8
	E_IPC	259.7	427.4	511.4	773.4	1033.8
Up ratio(%)		19.88	18.69	17.21	11.58	7.93

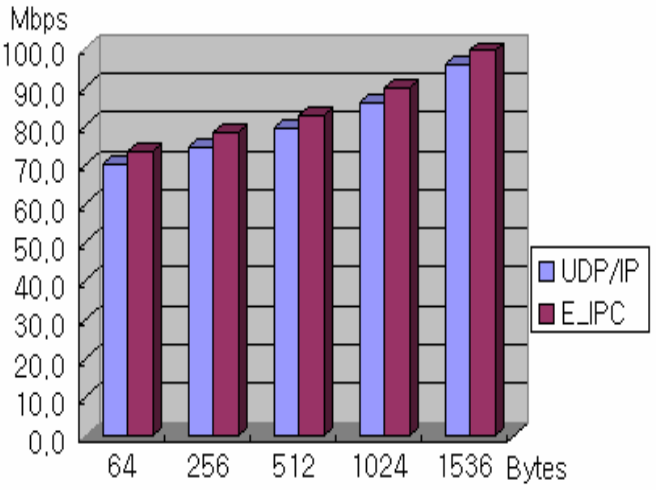


(그림 7) Round trip 시간 시험 결과 그래프

2) Throughput 시험

그림 8 은 호스트 A 와 호스트 B 사이에 UDP/IP 를 통한 Throughput 과 E_IPC 를 이용한 Throughput 의 측정 결과를 보여준다. 그림에서 알 수 있듯이 패킷 처리 능력에서도 E_IPC 에서 평균 4.3%정도 개선되었음을 확인할 수 있다. 패킷의 크기가 64 바이트일 경우 4.3%, 256 바이트일 경우 4.6%, 512 바이트일 경우 4.4%, 1024 바이트일 경우 4.1%, 1536 바이트일 경우 4.0%의 성능 개선효과를 보여준다.

발생시킨 데이터의 량은 일정한데 비해 패킷의 크기가 작을수록 Throughput 이 떨어지는 것은 패킷의 크기가 작을수록 프로토콜 계층을 지나면서 더 많은 Encapsulation 과 Decapsulation 을 해야 되므로 처리속도가 상대적으로 떨어지는 현상이 발생하기 때문이다.



(그림 8) Throughput 시험 결과 그래프

Prototype 시스템을 이용한 시험 결과 본 논문에서 제안한 E_IPC 를 이용한 통신 방식이 UDP/IP 를 이용한 경우 보다 RTT 와 Throughput 이 약 15.1%, 4.3%가 개선되었다.

4. 결론

오늘날 인터넷 사용자의 급속한 확산과 대용량 멀티미디어 데이터의 폭발적인 증가에 비해 물리적 네트워크 시스템의 성능 개선은 수요자의 고 품질 및 고 대역폭 요구를 충족시키지 못하고 있다. 이는 결과적으로 라우터 시스템의 병목현상을 발생시키고 네트워크의 성능을 악화시키고 있다.

이러한 문제점을 개선하기 위해, 비교적 신속한 처리가 요구되는 라우터의 관리 및 제어 관련된 메시지를 전송하는 기존의 Ethernet IPC 방식을 개선하여 E_IPC 방식을 제안하였다.

제안한 E_IPC 방식을 구현하여 prototype 시스템에 적용한 결과, 관리 및 제어와 관련된 메시지는 3 계층을 통하지 않고 1,2 계층에서 처리토록 하여 데이터 전송 속도가 약 15.1% 개선되었으며 패킷 처리능력은 4.3% 개선되어 라우터 시스템의 성능향상에 기여하였다.

참고문헌

[1] S.Keshav, R.Sharma, "Issues and trends in router design," IEEE Communications Magazine,Vol.36, pp.144-151, May 1998.

[2] D.Decasper, Z.Dittia and Guru Parulkar, "Router Plugins: A Software Architecure for Next-Generation Routers," IEEE/ACM Trans. on Networking, Vol 8. No.1, Feb 2000.

[3] 김덕기, 이상우, 이준철 “고속 네트워크 시스템의 이중화 회로 구현”, 대한전자공학회추계종합학술대회논문집, Vol.23, No.2, pp.267-270, 2000.

[4] V.P. Kumar, T.V. Lashman, D. Stiliadis, "Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet," IEEE Communications Magazine, Vol.36, pp.152-164, May 1998.

[5] 성정식, 허재두, 이형호 "메트로망에서의 IP 데이터 트래픽 전송 기술에 관한 고찰", ETRI 전자통신동향분석 통권 77 호 제 17 권 제 5 호, 2002.

[6] Wang-Bong Lee, etal, "An Architecture of Distributed Multi-Gigabit IP Router," AIC 24th Conference, Seoul, Nov. 2000.

[7] Bup Joong Kim, et al, "Design and Implementation of IPC Network in ATM Switching System,"ICATM, pp.148-152, 2001.

[8] 김성혜, 이규호 “대형 라우터 시스템에서 분산된 데이터 전달 기능에 대한 연구”, 한국정보과학회가을학술발표논문집, Vol.28, No.2, pp.247-250, 2001.

[9] J. Furnuas, et al, "A prototype for interprocess communication support, in hardware," Ninth Euromicro Workshop on Real-Time Systems, pp.18-24, 1997.

[10] Daniel Sutcliffe, Geoff Hall, Olaf Kock, Richard Clarke, "OpenSTA1.4.2," http://sourceforge.net/projects/opensta/, Sep. 2000.