

Hybrid TCP/IP Offload Engine의 프로토타입 개발*

장한국^o 박종훈 정상화
부산대학교 컴퓨터공학과
{hkjang^o, hanul, shchung}@pusan.ac.kr

Development of a Prototype for Hybrid TCP/IP Offload Engine

Hankook Jang^o Jong-Hoon Park Sang-Hwa Chung
Department of Computer Engineering, Pusan National University

요 약

TOE(TCP/IP Offload Engine)은 호스트 CPU가 아닌 네트워크 어댑터 상에서 TCP/IP 프로토콜을 처리하여 호스트 CPU의 부하를 줄이는 기술이다. TOE의 구현 방안으로는 임베디드 프로세서를 사용하여 TCP/IP를 처리하는 소프트웨어적인 접근 방법과 TCP/IP의 모든 기능을 하드웨어로 구현하는 접근 방법이 제안되어왔다. 본 논문에서는 하드웨어적인 접근 방법과 소프트웨어적인 접근 방법을 결합한 Hybrid TOE 구조를 개발하기 위해 FPGA와 ARM 프로세서에 기반한 프로토타입을 개발하였다. Hybrid TOE는 많은 작업 부하로 인하여 임베디드 프로세서 상에서 성능을 확보하기 어려운 기능들은 하드웨어로 구현하고, 연결 설정과 같이 통신의 성능에 큰 영향을 끼치지 않는 기능들은 임베디드 프로세서 상에서 소프트웨어로 처리한다. 또한 본 논문에서는 실험을 통해 Hybrid TOE 프로토타입이 호스트 CPU 상에 발생하는 부하를 줄임을 입증하고, 하드웨어 구현을 통해 통신의 성능을 향상시킬 수 있음을 보였다.

1. 서 론

현재 가장 널리 사용되고 있는 통신 프로토콜인 TCP/IP는 컴퓨터 시스템의 호스트 CPU가 처리하는 것이 일반적인데, 이는 호스트 CPU 상에 막대한 부하(load)를 발생시켜 전체 시스템의 성능을 저하시킨다는 문제점을 가진다[1]. 이를 해결하는 방안으로 최근 TCP/IP를 네트워크 어댑터에서 처리하는 TOE (TCP/IP Offload Engine) 기술이 제안되었다.

TOE를 개발하는 데에는 두 가지 접근 방법이 제안되어 왔다. 첫 번째 방법은 네트워크 어댑터에 탑재한 임베디드 프로세서 상에서 소프트웨어로 TCP/IP를 처리하는 방안으로, TCP/IP를 하드웨어로 구현하는 방안에 비해 구현이 쉽다는 장점을 가진다 [2]. 그러나 임베디드 프로세서는 호스트 CPU에 비해 성능이 낮으므로, 네트워크의 성능이 다소 떨어지는 단점이 있다. 두 번째 방법은 TCP/IP를 처리하는 전용 ASIC을 개발하는 방안이다 [3][4]. 이 방안은 네트워크의 성능을 보장할 수 있지만 [5], TCP/IP를 기반으로 하는 상위 수준 프로토콜을 네트워크 카드에서 같이 처리하고자 할 때 이에 대응하기가 어렵다.

본 연구진은 선행 연구[6]에서 하드웨어 기반의 TOE 구현 방안과 소프트웨어 기반의 TOE 구현 방안을 결합한 Hybrid TOE 구조를 제안하였다. Hybrid TOE 구조는 많은 작업 부하로 인하여 임베디드 프로세서 상에서 성능을 확보하기 어려운 기능들은 하드웨어로 구현하고, 연결 설정과 같이 통신

의 성능에 영향을 크게 끼치지 않는 기능들은 임베디드 프로세서 상에서 소프트웨어로 처리한다. 그 결과 Hybrid TOE는 하드웨어 기반의 TOE에 근접하는 성능을 제공할 수 있으며, 새로운 기능을 추가하거나 TCP/IP를 기반으로 하는 상위 프로토콜까지 네트워크 어댑터에서 처리할 수 있으므로 구조의 유연성 측면에서 장점을 가진다.

본 논문에서는 FPGA와 ARM 프로세서를 탑재한 보드를 개발하고, 이를 기반으로 Hybrid TOE의 프로토타입을 개발하였다. Hybrid TOE 프로토타입의 하드웨어 모듈은 FPGA 상에 구현되고, 소프트웨어 모듈은 ARM 프로세서 상에서 구현된다. 본 논문에서는 실험을 통해 Hybrid TOE가 호스트 CPU 상에서 발생하는 부하를 크게 감소시킬 수 있음을 입증하였고, 또한 하드웨어 구현을 통해 통신 성능의 향상이 가능함을 입증하였다.

본 논문은 구성은 다음과 같다. 2장에서는 Hybrid TOE 구조에 대해 설명하고, 3장에서 Hybrid TOE 프로토타입에 대해 설명한다. 4장에서는 실험 결과를 제시한다. 마지막으로 5장에서는 결론과 향후 연구를 제시한다.

2. Hybrid TOE 구조

그림 1은 Hybrid TOE와 이에 기반한 네트워크 어댑터의 구조를 보여준다. Hybrid TOE 어댑터는 Hybrid TOE Module, TOE Interface, Memory Controller 및 Gigabit Ethernet Controller로 구성된다. Hybrid TOE Module은 Hybrid TOE 어댑터의 핵심 모듈로서, 두 개의 임베디드 프로세서 코어와 한 개의 TOE Hardware Module로 구성된다.

* 이 논문은 교육인적자원부 지방연구중심대학육성사업(차세대 물류IT기술연구사업단)의 지원에 의하여 연구되었음.

임베디드 프로세서는 소프트웨어로 처리할 TCP/IP 기능을 구현하며, 두 개의 임베디드 프로세서를 사용하여 송수신과정을 분담하여 처리함으로써 호스트 CPU에 비해 성능이 떨어지는 임베디드 프로세서의 단점을 극복할 수 있다. 또한 송신 프로세스와 수신 프로세스 사이의 스케줄링에 의한 작업 전환 오버헤드를 제거할 수 있다. TOE Hardware Module은 하드웨어로 처리할 TCP/IP 기능을 구현하며, TOE Interface는 호스트 CPU와의 인터페이스를 담당한다. Memory Controller는 패킷 버퍼의 운용에 필요한 버퍼 메모리를 관리하는 역할을 담당한다. 패킷 버퍼는 TCP, IP, 그리고 MAC 헤더들을 저장하기 위한 헤더 영역과 송수신 데이터를 저장하기 위한 데이터 영역으로 구성되며, 송수신 패킷을 저장하는 데 사용된다. Gigabit Ethernet Controller는 Gigabit Ethernet MAC/PHY 칩과의 인터페이스를 담당한다.

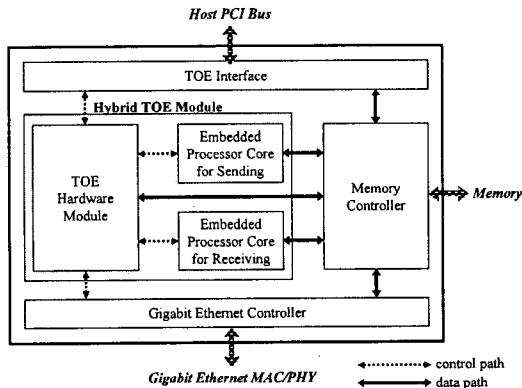


그림 1. Hybrid TOE 어댑터의 구조

Hybrid TOE 어댑터를 사용한 통신의 전체적인 동작 과정은 다음과 같다. 호스트 CPU의 사용자 프로그램이 Hybrid TOE를 사용한 통신을 요청하면, 이 요청은 호스트의 운영체제의 TCP/IP 프로토콜 스택을 거치지 않고 TOE Interface로 직접 전달된다. TOE Interface는 이 요청을 Hybrid TOE Module로 전달하고, Hybrid TOE Module에서는 하드웨어와 소프트웨어를 연동하여 요청된 작업을 처리한다. 요청된 작업이 데이터의 송신인 경우 Hybrid TOE Module은 패킷 버퍼를 생성한 후 원격 노드로 전송할 데이터를 호스트 CPU의 메인 메모리에서 패킷 버퍼의 데이터 영역으로 DMA를 사용하여 가져온다. Hybrid TOE Module은 패킷 버퍼의 헤더 영역에 TCP/IP 헤더와 MAC 헤더를 생성하여 패킷을 완성하고, 패킷 생성이 끝나면 Gigabit Ethernet Controller에 패킷의 전송을 요청한다. 마지막으로 Gigabit Ethernet Controller의 요청을 받은 Gigabit Ethernet MAC이 DMA를 사용하여 패킷 버퍼로부터 그 패킷을 가져가서 수신 노드로 전송한다.

수신 과정에서는 송신 과정의 역순으로 수신 패킷이 처리된다. 요청된 작업의 처리가 끝나면 TOE Interface가 호스트 CPU에 인터럽트를 발생시켜 처리가 완료되었음을 알리고, 호스트 CPU는 TOE Interface에 저장된 처리 결과를 읽어가서 이를 사용자 프로그램에 반환한다.

3. Hybrid TOE 프로토타입

본 논문에서는 Hybrid TOE의 최종 구현에 앞서 FPGA와

단일 임베디드 프로세서에 기반한 Hybrid TOE 프로토타입을 개발하였다. 그림 2는 Hybrid TOE 프로토타입 보드의 구조를 보여준다. FPGA는 Xilinx사의 Virtex-II Pro (XC2VP30)를 채택하였고, 임베디드 프로세서는 삼성전자의 ARM 프로세서(S3C2410X)를 채택하였다. 그리고, Gigabit Ethernet MAC/PHY chipset으로는 National사의 DP82820/DP83865을 사용하였다. FPGA에는 TOE Interface, TOE Hardware Module, HW/SW Interface 및 Gigabit Ethernet Controller가 구현되었다. 이 중에서 TOE Hardware Module은 송신 과정에서 패킷 버퍼와 관련된 세 가지 기능을 처리한다. 그리고, HW/SW Interface는 하드웨어와 소프트웨어 사이의 연동 메커니즘을 구성하는 핵심 요소로서 TOE Hardware Module과 ARM 프로세서 사이의 인터페이스를 제공한다.

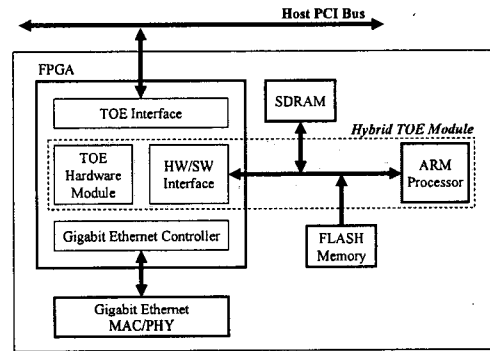


그림 2. Hybrid TOE 프로토타입 보드

Hybrid TOE 프로토타입에서는 그림 3과 같이 송신 과정에서 패킷 버퍼의 생성과 사용에 관련된 세 가지의 하드웨어 유닛과 함께 하드웨어 모듈과 소프트웨어 모듈의 연동을 지원하는 HW/SW Interface를 개발하였다.

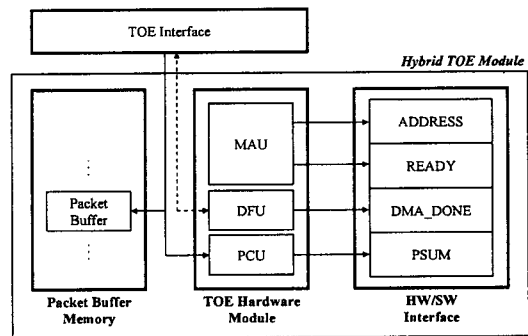


그림 3. TOE Hardware Module과 HW/SW Interface

TOE Hardware Module에서 MAU (Memory Allocation Unit)는 데이터 및 헤더를 저장하는 패킷 버퍼를 생성하기 위해 패킷 버퍼 메모리를 관리한다. DFU (Data Fetch Unit)는 TOE Interface 내부의 DMA 엔진을 사용하여 메인 메모리에서 패킷 버퍼의 데이터 영역으로 송신 데이터를 복사한다. PCU (Partial Checksum Calculation Unit)는 DFU에 의해 데이터가 복사되는 동안 데이터의 부분(partial) 체크섬을 계산한다.

HW/SW Interface에서 ADDRESS 필드는 MAU에 의해 할당된 패킷 버퍼의 주소를 저장하고, READY 필드는 패킷 버퍼가 준비되었는지의 여부를 나타낸다. DMA_DONE 필드는 DFU에 의한 데이터 복사가 끝났는지의 여부를 나타내고, PSUM 필드에는 PCU가 계산한 부분 체크섬이 저장된다.

Hybrid TOE 프로토타입의 소프트웨어 모듈은 ARM 프로세서 상에서 임베디드 리눅스를 사용하여 개발하였다. 임베디드 리눅스는 다른 운영체제에 비해서 네트워크 성능이 우수하고, TCP/IP 프로토콜 스택을 포함한 커널의 수정이 용이하기 때문에 선택되었다. 또한 임베디드 리눅스를 사용하면 TCP/IP를 기반으로 하는 상위 프로토콜을 네트워크 어댑터에서 오프로딩하는 방향으로 TOE를 응용할 때 편리한 개발 환경을 제공할 수 있다.

ARM 프로세서는 소켓 버퍼 구조체를 생성할 때 READY 필드를 폴링하여 패킷 버퍼가 생성되었는지를 검사한다. READY 필드가 세팅되면 ARM 프로세서는 ADDRESS 필드를 통해 패킷 버퍼의 주소를 획득한다. 이어서 ARM 프로세서는 DMA_DONE 필드를 폴링하여 데이터 복사 및 부분 체크섬 계산의 완료를 감시하고, DMA_DONE 필드가 세팅되면 PSUM 필드에서 부분 체크섬을 가져온다. 이후 ARM 프로세서는 TCP/IP 헤더 및 MAC 헤더를 생성하여 패킷 버퍼의 헤더 영역으로 복사하고, 생성된 패킷은 수신 노드로 전송된다.

4. 실험 및 분석

그림 4는 TCP/IP 기반 통신을 수행할 때 Hybrid TOE 프로토타입(Hybrid TOE)을 사용한 경우와 일반 Gigabit Ethernet 어댑터(GBE)를 사용한 경우의 호스트 CPU 점유율을 비교하고 있다. Gigabit Ethernet 어댑터로는 Intel PRO/1000MT Server Adapter를 사용하였다. 일반 Gigabit Ethernet 어댑터를 장착한 시스템의 경우 호스트 CPU 점유율은 데이터의 크기에 따라 50~80% 정도인 반면에, Hybrid TOE 프로토타입을 장착한 시스템의 경우 호스트 CPU 점유율은 약 9%로 일정하며 일반 Gigabit Ethernet 어댑터를 장착한 시스템보다 훨씬 낮은 것을 알 수 있다. 이러한 결과는 Hybrid TOE 프로토타입이 호스트 CPU에서 TCP/IP를 처리하는 부하를 크게 감소시키는 효과를 입증하는 것이다.

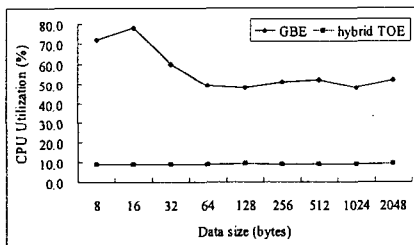


그림 4. TOE와 일반 NIC의 CPU 점유율 비교

표 1은 임베디드 리눅스만을 사용하여 구현한 소프트웨어 TOE(S/W TOE)와 본 논문에서 개발한 하드웨어 유닛을 적용한 Hybrid TOE 프로토타입(Hybrid TOE)의 성능을 비교한 결과로서, 하드웨어 유닛을 적용한 경우(S2, S3, S8) 소프트웨어 구현에 비해 약 9~56배의 성능 향상이 있음을 알 수 있다.

표 1. Hybrid TOE와 S/W TOE의 성능 비교 (단위 : μ s)

Label	Operation	Hybrid TOE	S/W TOE
S1	Enter TCP layer from interrupt handler	469.2	469.2
S2	Create packet buffer	1.0	10.0
S3	Fetch data from main memory	7.7	69.6
S4	Create TCP header	48.9	48.9
S5	Enter IP layer from TCP layer	52.6	52.6
S6	Create IP header	49.0	49.0
S7	Enter device driver from IP layer and Create MAC header	32.7	32.7
S8	Copy packet to TX Buffer	12.2	690.5
S9	Complete transmission	104.2	104.2

표 1의 결과에서 보듯이 소프트웨어로 TCP/IP를 처리할 때 TCP/IP/MAC 헤더를 생성하는 과정, 각 계층간의 전환 과정, 처리 결과를 생성하여 호스트 CPU로 보고하는 과정 등에서 많은 시간이 소모되고 있다. 따라서, Hybrid TOE의 성능을 극대화하기 위해서는 각 계층간의 전환 과정에서 발생하는 오버헤드를 줄이고, 헤더 생성 및 처리 결과의 보고를 하드웨어로 처리하는 등의 최적화가 필요할 것으로 분석된다.

5. 결론 및 향후 연구

본 논문에서는 FPGA와 ARM 프로세서를 기반으로 하는 Hybrid TOE 프로토타입을 개발하였다. 실험 결과 Hybrid TOE 프로토타입의 CPU 점유율은 9% 이하로서 호스트 CPU에서 발생하는 부하를 크게 감소시키는 효과를 입증하였다. 또한, 하드웨어/소프트웨어를 연동한 Hybrid TOE 프로토타입이 소프트웨어 TOE에 비해서 성능이 우수하였다. 향후 과제로는 헤더 생성 등의 과정을 하드웨어로 처리하는 등의 최적화를 통해 Hybrid TOE의 성능을 극대화할 계획이다.

6. 참고 문헌

- [1] E. Yeh, H. Chao, V. Mannem, J. Gervais and B. Booth, "Introduction to TCP/IP Offload Engine (TOE)", 10 Gigabit Ethernet Alliance, April 2002
- [2] Intel PRO/1000T IP Storage Adapter, <http://www.intel.com>
- [3] SLIC Technology, <http://www.alacritech.com>
- [4] NAC-7711, <http://www.adaptec.com>
- [5] H. Ghadia, "Benefits of full TCP/IP offload (TOE) for NFS Services", Proceedings of 2003 NFS Industry Conference, <http://nfsconf.com/pres03/adaptec.pdf>, Sep. 2003
- [6] S. C. Oh, H. Jang and S. H. Chung, "Analysis of TCP/IP protocol stack for a Hybrid TCP/IP Offload Engine", Proceedings of The 5th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'04), pp. 406-409, Dec. 2004