하이브리드 TCP/IP Offload Engine을 위한 하드웨어 기반 송수신 가속기의 설계 및 구현

(Design and Implementation of a Hardware-based Transmission/Reception Accelerator for a Hybrid TCP/IP Offload Engine)

장 한 국 † 정 상 화 †† 유 대 현 †
(Han-Kook Jang) (Sang-Hwa Chung) (Dae-Hyun Yoo)

요 약 최근 Gbps 이상의 고속 네트워크 상에서 호스트 CPU에 많은 오버헤드를 발생시키는 TCP/IP의 문제점을 해결하기 위해 네트워크 어댑터 상에서 TCP/IP를 처리함으로써 호스트 CPU의 작업부하를 줄이는 TCP/IP Offload Engine(TOE) 기술이 연구되고 있다. TOE의 구현 방법에는 범용 임베디드 프로세서에서 소프트웨어로 TCP/IP를 처리하는 방법과 전용 ASIC에서 하드웨어로 TCP/IP를 처리하는 방법이 사용되어 왔으나 소프트웨어 구현은 통신의 성능이 떨어지고 하드웨어 구현은 유연성과 확장성이 떨어지는 문제점들을 가지고 있다. 본 논문에서는 하드웨어적인 접근 방법과 소프트웨어적인 접근 방법을 결합한 하이브리드 TOE 구조를 제안한다. 하이브리드 TOE는 데이타 패킷의 생성과 처리와 같이 통신의 성능에 큰 영향을 끼치는 기능들을 하드웨어로 구현함으로써 하드웨어 기반 TOE 구현에 버금가는 성능을 제공하고, 연결 설정과 같이 통신의 성능에 영향을 크게 끼치지 않는 기능들은 임베디드 프로세서 상에서 소프트웨어로 처리한다. 본 논문에서는 데이타 송수신의 성능을 높이기 위해 데이타 패킷의 생성 및 처리 등을 지원하는 하드웨어 송수신 가속기를 설계 및 구현하였다. 실험 결과 송수신 가속기를 사용한 하이브리드 TOE는 약 19 μs의 최소 지연시간을 보였다. 그리고 6% 이하의 CPU 점유율에서 약 675 Mbps에 달하는 대역폭을 보였다.

키워드 : TCP/IP, TCP/IP Offload Engine, 하이브리드 TOE, 송수신 가속기

Abstract TCP/IP processing imposes a heavy load on the host CPU when it is processed by the host CPU on a very high-speed network. Recently the TCP/IP Offload Engine (TOE), which processes TCP/IP on a network adapter instead of the host CPU, has become an attractive solution to reduce the load in the host CPU. There have been two approaches to implement TOE. One is the software TOE in which TCP/IP is processed by an embedded processor and the other is the hardware TOE in which TCP/IP is processed by a dedicated ASIC. The software TOE has poor performance and the hardware TOE is neither flexible nor expandable enough to add new features. In this paper we designed and implemented a hybrid TOE architecture, in which TCP/IP is processed by cooperation of hardware and software, based on an FPGA that has two embedded processor cores. The hybrid TOE can have high performance by processing time-critical operations such as making and processing data packets in hardware. The software based on the embedded Linux performs operations that are not time-critical such as connection establishment, flow control and congestions, thus the hybrid TOE can have enough flexibility and expandability. To improve the performance of the hybrid TOE, we developed a hardware-based transmission/reception accelerator that processes important operations such as creating data packets. In the experiments the hybrid TOE shows the minimum latency

· 이 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (지방연구중심대학육성사업/차세대물류IT기술연구사업단)
† 학생회원 : 부산대학교 컴퓨터공학과
hkjang@pusan.ac.kr
yoodh@pusan.ac.kr
†† 종신회원 : 부산대학교 컴퓨터공학과 교수
shchung@pusan.ac.kr
(Corresponding author임)
논문접수 : 2007년 6월 13일
심사완료 : 2007년 7월 22일

of about 19 μs. The CPU utilization of the hybrid TOE is below 6 % and the maximum bandwidth of the hybrid TOE is about 675 Mbps.

Key words : TCP/IP, TCP/IP Offload Engine, Hybrid TOE, Transmission/Reception Accelerator

1. 서 론

현재 TCP/IP 프로토콜은 일반적으로 운영체제에 포함되어 호스트 CPU 상에서 처리되고 있다. 이렇게 호스트 CPU 상에서 TCP/IP를 처리하는 방식은 Gbps 이상의 속도를 제공하는 초고속 네트워크 환경에서는 호스트 CPU 상에 막대한 부하(load)를 유발하여 전체 시스템의 성능을 저하시킨다는 문제점을 가진다[1]. 또한 1 bps 속도로 TCP/IP를 처리하는 데 약 1 Hz의 CPU 클럭이 필요하므로 네트워크의 물리적인 대역폭이 10 Gbps 이상으로 높아지면 단일 호스트 CPU로는 TCP/IP를 처리하기 어려워지는 상황이 발생할 것으로 예상되고 있다. 이러한 문제점을 해결하는 방안으로서 호스트 CPU가 아닌 네트워크 어댑터에서 TCP/IP를 처리하는 TCP/IP Offload Engine(TOE) 기술이 제안되고 있다 [2]. TCP/IP가 네트워크 어댑터 상에서 처리되면 호스트 CPU의 입장에서는 프로토콜을 처리하는 작업 부하가 줄어들어 실질적인 작업에 더 많은 CPU 자원을 할당할 수 있게 된다. 이는 컴퓨터 시스템의 전체적인 성능이 향상되는 효과로 이어진다.

TOE를 구현하는 방식으로는 네트워크 어댑터에 탑재한 범용 임베디드 프로세서 상에서 소프트웨어로 TCP/IP를 처리하는 소프트웨어 TOE[3,4], 전용 ASIC을 개발하여 하드웨어로 TCP/IP를 처리하는 하드웨어 TOE [5-9], 그리고 TCP/IP 기능 중 일부는 소프트웨어로 처리하고 일부는 하드웨어로 처리하는 하이브리드 TOE [10-12] 등이 제안되고 있다. 소프트웨어 TOE는 하드웨어 TOE에 비해 구현이 쉽다는 장점을 가진다. 그러나 임베디드 프로세서는 일반적으로 호스트 CPU에 비해 성능이 낮으므로 하드웨어 TOE에 비해 프로토콜을 처리하는 성능이 떨어진다[13]. 하드웨어 TOE는 소프트웨어 TOE에 비해 통신의 성능은 우수하지만[14-16], 구현이 비교적 어렵고 개발에 비용이 많이 든다. 또한 TCP/IP의 상위 프로토콜까지 네트워크 어댑터 상에서 처리하려고 할 때 상위 프로토콜을 처리하는 하드웨어 모듈을 새로 개발하여 기존 TCP/IP 처리 모듈과 결합해야 하는 어려움이 발생한다. 하드웨어 TOE와 소프트웨어 TOE를 결합한 하이브리드 TOE는 데이타 패킷의 생성과 처리와 같이 많은 작업 부하로 인하여 임베디드 프로세서 상에서 성능을 확보하기 어려운 기능들은 하드웨어로 구현함으로써 하드웨어 TOE에 근접하는 성능을 제공한다. 그리고 연결 설정과 같이 통신의 성능에 영향을 크게 끼치지 않는 기능들은 임베디드 프로세서 상에서 소프트웨어로 처리함으로써 하드웨어 TOE에 비해 비교적 구현이 쉽다는 장점을 가진다. 또한 차후 TCP/IP의 상위 프로토콜을 오프로딩하거나 새로운 기능을 추가하기가 용이하다.

본 논문에서는 두 개의 프로세서 코어를 내장한 FPGA 상에서 하이브리드 TOE의 데이타 송수신 과정에 필요한 주요 기능들을 처리하는 하드웨어 기반의 송수신 가속기를 설계 및 구현하였다. 그리고 하드웨어 송수신 가속기를 구성하는 모듈들과 프로세서 코어들의 연동을 지원하는 메커니즘을 개발하였다. 하드웨어 모듈들은 데이타 패킷 헤더의 생성 및 처리, DMA를 사용한 데이타 수집 및 저장 등을 담당하여 송수신 성능을 향상시킨다. 하이브리드 TOE에서 데이타 송수신 이외의 기능들은 임베디드 리눅스 기반의 소프트웨어를 바탕으로 처리하도록 구현하였다. 두 개의 프로세서 코어들은 각각 송신 하드웨어 모듈과 수신 하드웨어 모듈과 결합하여 송신 기능과 수신 기능을 협력하여 처리한다. 본 논문에서는 프로세서 코어 내장형 FPGA가 장착된 TOE 네트워크 어댑터를 사용하여 하이브리드 TOE의 동작을 검증하였고, 실험을 통해 하드웨어 기반 송수신 가속기의 성능을 입증하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구를 소개하고, 3장에서는 하이브리드 TOE의 구조와 하드웨어 기반 송수신 가속기 및 소프트웨어 모듈의 구현에 대해 설명한다. 4장에서는 실험 결과를 제시한다. 마지막으로 5장에서는 결론과 향후 연구를 제시한다.

2. 관련연구

TOE 관련된 제품의 개발 사례 중에서 범용 임베디드 프로세서를 사용하여 소프트웨어 TOE를 구현한 사례로는 Intel사의 PRO/1000T IP Storage Adapter[3]와 Hewlett Packard사의 시제품[4] 등이 있었다. Intel의 제품은 하드웨어 TOE에 비해 성능이 크게 떨어지고 [13], HP의 제품은 시제품으로 끝나서 현재 제품의 생산이 이루어지지 않고 있다.

1000Base-T 규격의 Gigabit Ethernet와 ASIC 구현 기반의 하드웨어 TOE 제품에는 Alacritech사의 SLIC technology[5], QLogic사의 QLA4050C 어댑터[6], Broadcom사의 BCM5706 controller[7] 등이 있다. ASIC 구현과 10 Gigabit Ethernet에 기반한 하드웨어

TOE의 구현 사례로는 Chelsio사의 Terminator 3 칩[8], NetEffect사의 NE010 어댑터[9] 등이 있다. 각종 자료에 의하면 하드웨어 기반 TOE 제품들은 대체로 100 MB/s 정도의 높은 단방향 대역폭을 보유하고 있다[13-15]. 그러나 하드웨어 TOE 구현은 전용 ASIC 구현에 많은 시간과 비용이 소모되고 구현된 하드웨어에서 수정하거나 개선할 사항이 발생할 때마다 새로운 ASIC을 개발해야 한다는 단점을 가진다. 또한 RDMA (Remote Direct Memory Access) 등과 같은 상위 프로토콜까지 네트워크 어댑터에서 오프로딩하여 처리하려는 요구가 발생할 때에도 효과적으로 대응하기 어렵다.

근래에는 FPGA와 임베디드 프로세서를 기반으로 하여 하이브리드 TOE 구현을 연구한 논문들이 발표되고 있다[10-12]. 하이브리드 방식의 TOE 구현에 대해 연구한 Wu와 Chen의 논문에서는 IP, ARP, ICMP 프로토콜들을 하드웨어로 처리하고 TCP는 1개의 PowerPC 코어를 사용하여 소프트웨어로 처리하는 구조를 가지고 있다[12]. 이 연구의 결과를 보면 통신 성능이 TCP까지 하드웨어로 처리하는 우리의 구현보다 크게 떨어진다.

3. 송수신 가속기에 기반한 하이브리드 TOE 구현

본 장에서는 두 개의 임베디드 프로세서 코어와 하드웨어 기반 송수신 가속기를 결합한 하이브리드 TOE에서 하드웨어 구현과 소프트웨어 구현에 대해 설명한다. 그림 1은 듀얼 프로세서 코어 내장형 FPGA와 이에 기반한 하이브리드 TOE 네트워크 어댑터의 구조를 나타낸다. 본 논문에서는 두 개의 PowerPC 405 코어를 내장한 Xilinx사의 Virtex-II Pro FPGA를 사용하여 하이브리드 TOE를 구현하였다. 두 개의 프로세서 코어들은 각각 송신 처리를 분담하는 TX 프로세서와 수신 처리를 분담하는 RX 프로세서로 사용되며, 프로세서간의 인터페이스를 사용하여 상대방 프로세서에게 작업을 요청할 수 있다. 이렇게 두 프로세서 코어가 송수신 과정을 분담하여 처리하는 메커니즘은 송신 프로세스와 수신 프로세스 사이의 스케줄링에 의한 작업 전환 오버헤드를 제거하여 호스트 CPU에 비해 성능이 떨어지는 임베디드 프로세서의 단점을 극복할 수 있다. 각 코어들은 PLB(Processor Local Bus) 버스를 통해 FPGA 외부에 연결된 64MB 용량의 SDRAM과 연결되고, OPB (On-chip Peripheral Bus) 버스를 통해 32MB 용량의 플래시 메모리에 연결된다. 플래시 메모리에는 각 프로세서에서 운용될 소프트웨어들의 압축 이미지가 저장되고, SDRAM에서는 실제 소프트웨어가 운용된다. 각 코어들은 300 MHz의 코어 클럭, 100 MHz의 PLB 클럭, 50 MHz의 OPB 클럭으로 동작한다.

FPGA 내부의 하드웨어 모듈들은 호스트 인터페이스, TOE 모듈, GbE(Gigabit Ethernet) 컨트롤러 등으로 구성된다. 호스트 인터페이스는 64bit/66MHz PCI 컨트롤러를 내장하여 호스트 CPU와 TOE 모듈 사이에서 인터페이스를 담당한다. GbE 컨트롤러는 기가빗 이더넷 MAC/PHY 칩과의 인터페이스를 담당하여 이더넷 패킷의 송신과 수신을 처리한다. TOE 모듈은 하이브리드 TOE에서 하드웨어 구현의 핵심으로 데이타 송수신 기능 등을 처리하는 하드웨어 가속 모듈과 프로세서 코어들과의 인터페이스를 담당하는 HW/SW 인터페이스로

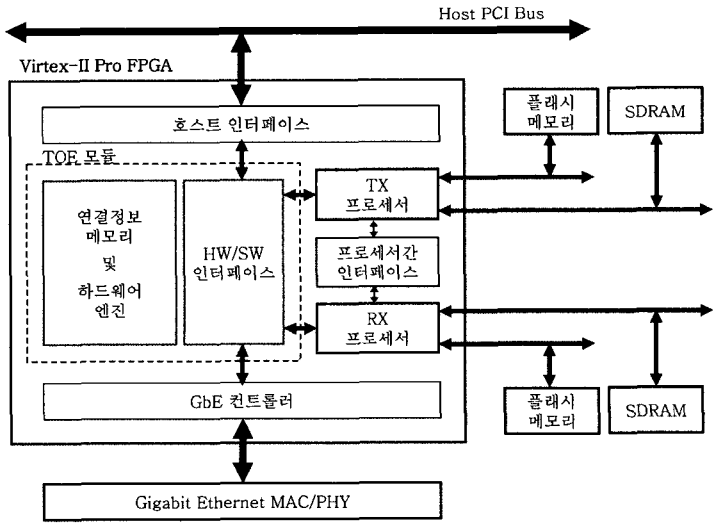

그림 1 하이브리드 TOE의 구조

구성된다. TOE 모듈은 TX 프로세서와 RX 프로세서의 PLB 버스에 양쪽으로 연결되며, TX 프로세서와 RX 프로세서는 메모리에 접근하는 방식으로 TOE 모듈에 접근할 수 있다. 송신 파트(TX 프로세서 및 송신 하드웨어 모듈)와 수신 파트(RX 프로세서 및 수신 하드웨어 모듈)는 TOE 모듈 내의 연결 정보 버퍼를 통해 TCP 연결(connection)에 대한 정보를 공유한다. TCP 연결 정보는 TCP, IP, MAC 헤더를 생성하거나 처리하는 데 필요한 정보들을 유지하며, 본 논문에서는 TCP 연결 정보를 저장하는 메모리 버퍼 블록에 대해 Connection Control Block(이하 CCB)이라는 이름을 사용한다.

임베디드 리눅스 기반의 소프트웨어는 하드웨어의 동작들을 제어하고 하드웨어로 구현되지 않은 기능들, 즉 연결 설정, ARP/ICMP 처리, 흐름 제어, 혼잡 제어, 재전송 등 데이타 송수신의 성능에 영향을 크게 끼치지 않는 나머지 동작들을 처리한다. 임베디드 리눅스를 사용함으로써 차후 패킷 필터링 등의 보안 함수나 상위 프로토콜의 오프로딩 등 새로운 기능들을 소프트웨어에 추가하기가 쉬우므로 본 논문의 하이브리드 TOE 구조는 높은 유연성과 확장성을 가진다.

3.1 하드웨어 송수신 가속기의 구조

그림 2는 하이브리드 TOE를 위한 송수신 가속기를 구성하는 하드웨어 모듈들의 구조를 보여준다. 송수신 처리를 위한 하드웨어 모듈은 크게 송신 버퍼 관리 모듈, 수신 버퍼 관리 모듈, 헤더 생성 모듈, 수신 처리 모듈, ACK 패킷 생성 모듈로 구성된다. 송신 버퍼는 GbE 컨트롤러로 전송할 패킷을 저장하는 역할을 한다. 프로세서에서 생성된 컨트롤 패킷, 헤더 생성 모듈에서 생성된 데이타 패킷, ACK 생성 모듈에서 생성된 ACK 패킷을 버퍼에 저장하고 GbE 컨트롤러로 전송하는 과정을 관리하는 것이 송신 버퍼 관리 모듈의 역할이다. 수신 버퍼는 GbE 컨트롤러에서 받은 패킷을 저장하는 역할을 한다. 수신 버퍼 관리 모듈은 수신 버퍼에 저장된 패킷에서 호스트 CPU의 메인 메모리로 데이타를 DMA 전송하는 역할을 한다. 헤더 생성 모듈은 데이타 패킷의 헤더를 생성하여 송신 버퍼에 저장되어있는 데이타 페이로드의 앞에 붙여 데이타 패킷을 완성하는 역할을 한다. 수신 처리 모듈은 수신 버퍼에 저장된 패킷을 검사하고 패킷을 분류하는 역할을 하며, 수신 패킷이 데이타 패킷인 경우 데이타를 분리하는 일을 맡는다. 마지막으로 ACK 패킷 생성 모듈은 수신된 데이타 패킷의 검사 결과 아무 이상이 없을 경우 ACK 패킷을 생성하여 송신 버퍼에 저장하는 역할을 한다.

TOE 네트워크 어댑터를 사용한 통신의 진행 과정은 다음과 같다. 호스트 CPU의 사용자 프로그램이 TOE를 사용한 통신을 요청하면 이 요청은 호스트 운영체제의 TCP/IP 프로토콜 스택을 거치지 않고 TOE 어댑터의 호스트 인터페이스에 직접 전달된다. 호스트 인터페이스는 이 요청을 TOE 모듈로 전달하고, TOE 모듈에서 하드웨어와 소프트웨어를 연동하여 요청된 작업을 처리한다. 연결 설정 작업은 TX, RX 프로세서의 역할이다. TX 프로세서가 연결 설정과 관련된 패킷을 송신 버퍼에 저장하고 전송 명령을 내리면 GbE 컨트롤러를 통해 전송된다. 반대로 수신 버퍼에 들어온 패킷이 연결 설정과 관련된 패킷이라면 수신 처리 모듈을 통해 이것을 확인하고 RX 프로세서에 인터럽트를 걸어서 처리하도록 한다.

요청된 작업이 데이타의 송신인 경우 TOE 모듈은 원격 노드로 전송할 데이타를 호스트 CPU의 메인 메모리에서 DMA 읽기를 사용하여 가져와서 송신 버퍼에 저장한다. 데이타 복사가 끝나면 헤더 생성 모듈을 통해서 TCP 헤더, IP헤더, MAC 헤더를 생성하고 각 헤더를

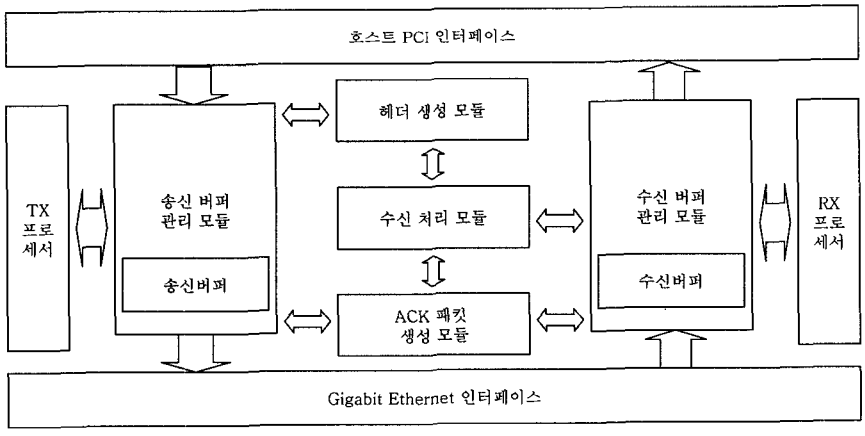

그림 2 하드웨어 기반 송수신 가속기의 구조

복사해온 데이타 페이로드의 앞에 붙인다. 패킷 생성이 끝나면 GbE 컨트롤러에 패킷의 전송을 요청한다. 기가빗 이더넷 MAC/PHY 칩에 의해 패킷 전송이 끝나면 호스트 CPU에 인터럽트를 걸어서 송신이 완료되었음을 알린다. 수신 과정에서는 송신 과정의 역순으로 수신 패킷이 처리된다. GbE 컨트롤러를 통해 패킷이 들어오면 수신 처리 모듈이 패킷을 검사한다. 패킷 검사를 통해 데이타 패킷이 확인되면 패킷에서 데이타를 분리해내고 DMA 쓰기를 통해서 호스트 메모리로 데이타를 복사한다. 데이타 복사가 끝나면 호스트 CPU에 인터럽트를 걸어서 데이타의 수신을 알린다.

3.2 송신 가속 모듈

송수신 가속을 위한 하드웨어 모듈은 헤더 생성 모듈, 수신 처리 모듈, ACK 패킷 생성 모듈로 구성된다. 헤더 생성 모듈은 데이타 패킷의 송수신시 하드웨어 모듈이 헤더를 만들고 패킷을 전송함으로써 송신 속도를 증가한다. 반대로 수신 패킷의 검사와 ACK패킷 전송을 각각 수신 처리 모듈과 ACK 패킷 생성 모듈이 처리하여 수신 속도를 증가한다.

그림 3은 송신 가속을 위한 헤더 생성 모듈의 내부 구조를 보여 준다. 헤더 생성 모듈은 헤더 정보 제어 모듈, 연결 정보 버퍼, CCB 캐시, 헤더 생성 제어 모듈, 송신버퍼 인터페이스로 구성된다. 연결 정보 버퍼는 헤더 생성을 위한 연결 정보를 저장하고 있는 역할을 맡는다. CCB 캐시는 헤더 생성에 사용할 연결 정보를 연결 정보 버퍼에서 읽어와 임시로 저장하는 역할을 맡는다. 연결 정보 버퍼와 CCB 캐시의 입출력 제어를 담당하는 것이 헤더 정보 제어 모듈의 역할이다. 헤더 생성 제어 모듈은 CCB 캐시로부터 연결정보를 읽어 헤더를 생성하는 역할을 한다. 헤더 생성이 끝나면 헤더를 송신 버퍼로 복사해야 하는데, 이 역할을 맡는 것이 송신 버퍼 인터페이스이다.

헤더 생성 모듈의 동작 과정은 다음과 같다. 송신 버퍼 관리 모듈에서 전송 데이타의 복사가 끝나면 데이타 체크섬, 길이 등의 정보와 전송 관련 정보가 헤더 생성 모듈로 전달된다. 헤더 정보 제어 모듈은 전송 관련 정보를 바탕으로 연결 정보 버퍼에서 관련 연결 정보를 찾고 이것을 CCB 캐시로 복사한다. 복사가 끝나면 헤더 생성 제어 모듈이 CCB 캐시와 데이타 체크섬, 길이 정보를 바탕으로 TCP 헤더, IP 헤더, MAC 헤더를 생성한다. 헤더 생성이 끝나면 송신버퍼 인터페이스를 통해서 송신버퍼로 전송 요청을 하고, 송신버퍼 관리 모듈이 허락하면 헤더를 송신버퍼로 복사한다.

3.3 수신 가속 모듈 및 ACK 처리 모듈

그림 4는 수신 처리 및 ACK 생성 모듈의 내부 구조를 보여준다. 수신 처리 및 ACK 생성 모듈은 수신 패킷 검사 모듈, 수신 패킷 헤더 레지스터 파일, 체크섬 계산 모듈, ACK 번호 계산 모듈, ACK 패킷 생성 모듈, 송신 버퍼 인터페이스로 구성된다. 수신 패킷 검사 모듈은 수신 패킷의 각각의 헤더 필드를 레지스터에 저장하고 패킷을 검사하고 처리하는 역할을 맡는다. 수신 패킷 헤더 레지스터 파일은 수신 패킷의 헤더 정보를 수신 처리 및 ACK 패킷 생성을 위해서 저장하고 있는 레지스터 파일이다. 체크섬 계산 모듈은 수신 패킷의 체크섬 검사와 ACK 패킷의 체크섬 생성하기 위해 TCP 체크섬과 IP 체크섬을 계산하는 역할을 맡는다. 수신 패킷의 데이타 길이 계산과 ACK 번호 계산 등은 ACK 번호 계산 모듈에서 처한다. ACK 패킷 생성 모듈은 수신 패킷 헤더 레지스터 파일과 체크섬 계산 모듈, ACK 번호 계산 모듈의 데이타를 바탕으로 ACK 패킷을 생성하는 역할을 한다. 생성된 ACK 패킷을 송신 버퍼로 전송하는 것은 송신 버퍼 인터페이스가 처리한다.

수신 처리 및 ACK 생성 모듈의 동작 과정은 다음과 같다. GbE 컨트롤러에서 수신 버퍼로 데이타가 들어오면 수신 패킷 검사 모듈은 각각의 헤더 필드를 분리하여 수신 패킷 헤더 레지스터 파일에 저장한다. 이때 체크섬 계산 모듈은 TCP 체크섬과 IP 체크섬을 계산한다. 수신 버퍼에 저장이 끝나면 체크섬 검사와 시퀀스

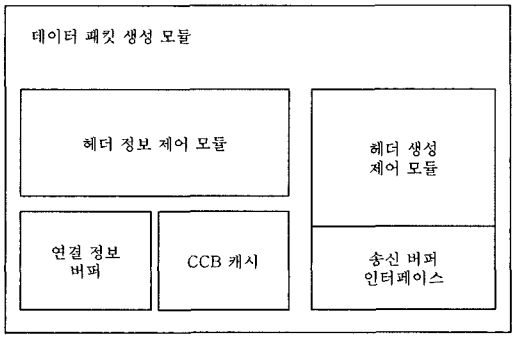

그림 3 송신 가속을 위한 하드웨어 모듈의 구조

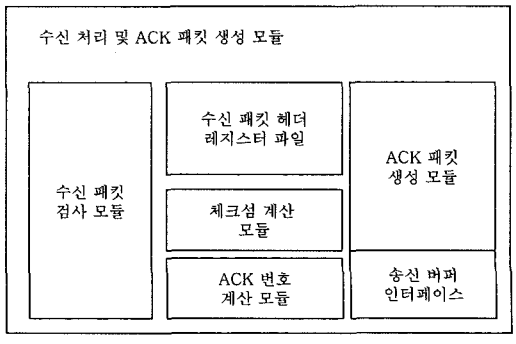

그림 4 수신 가속을 위한 하드웨어 모듈의 구조

번호 검사 등을 수행한다. 검사 결과를 바탕으로 ERROR, ARP, TCP_SYN(연결 설정을 위한 패킷), TCP_DATA (데이타 패킷)의 네 가지 패킷으로 분류하고 결과를 수신 버퍼 관리 모듈에게 알린다. 검사 결과에 따른 처리는 수신 버퍼 관리 모듈이 맡는다. 에러일 경우 수신 버퍼의 패킷을 버리고, ARP와 TCP_SYN 패킷은 RX 프로세서로 인터럽트를 걸어 처리한다. TCP_DATA 패킷이 수신된 경우 호스트 CPU에서 recv() call 명령이 내려왔는지를 먼저 확인한다. Recv() call 명령을 확인한 후에 패킷에서 데이타를 추출하여 호스트 메모리로 전송한다.

수신 버퍼 관리 모듈에서 수신 패킷의 처리가 진행되는 동안 ACK 패킷 생성 모듈에서는 ACK 패킷의 생성을 진행한다. 페이로드 길이 계산 결과로 ACK 번호를 계산하고, 수신 패킷 헤더 레지스터의 데이타를 바탕으로 ACK 패킷의 체크섬을 계산한다. ACK 번호 계산과 체크섬 계산이 끝나면 ACK 패킷을 생성하고, 송신 버퍼 인터페이스를 통해서 송신 버퍼에 저장한다. 저장된 ACK 패킷은 송신버퍼 관리 모듈에 의해서 GbE 컨트롤러를 통해 전송된다.

3.4 하이브리드 TOE를 위한 소프트웨어 모듈

본 구현에 사용되는 소프트웨어 모듈은 TCP/IP 프로토콜 스택을 내장한 임베디드 리눅스(커널 버전 2.4.18), 디바이스 드라이버, 응용 프로그램의 세 가지로 범주로 구성된다. TX 프로세서와 RX 프로세서는 각자 독자적인 SDRAM 메모리 상에서 임베디드 리눅스 커널과 디바이스 드라이버, 응용 프로그램을 운용한다. 송수신 경로의 분리를 위한 커널은 동일한 커널 이미지를 기본으로 하고, 각 프로세서의 역할에 따라 일정한 함수들만을 수행하도록 구현한다. 디바이스 드라이버와 응용 프로그램은 커널 이미지와 분리된 램 디스크 형태로 탑재되어 각 프로세서에 의해 수행된다. 커널 이미지와 램 디스크는 플래시 메모리에 압축된 형태로 저장되어 있다가 TOE 어댑터에 전원이 들어와 부팅이 시작될 때 압축이 풀리면서 SDRAM으로 복사된다. 본 논문에서 제안하는 송수신 분리형 하이브리드 TOE 구조에서는 TCP 연결에 대해 송신과 수신이 각각 다른 프로세서에 의해 관리되어야 하며, 두 프로세서들은 흐름 제어 등의 TCP 연결 관리를 위해 필요한 정보들을 공유해야 한다. 본 논문에서는 TX 프로세서, RX 프로세서, 하드웨어 모듈들 사이에서 연결 정보의 공유를 위해 하드웨어 모듈에서 제공하는 CCB (Connection Control Bolck) 단위의 연결 정보 버퍼를 사용한다.

4. 실험 및 분석

본 논문에서는 하이브리드 TOE의 성능을 실험하기 위해 AMD Opteron 246 CPU와 1 GB의 메모리를 장착한 2 대의 컴퓨터를 사용하였으며, 각 컴퓨터에는 2.6.19 커널 버전의 x86-64 아키텍처용 리눅스를 운영체제로 사용하였다. 리눅스 커널 설정에서 타이머의 주파수 해상도(timer frequency resolution)은 1,000 Hz로 설정하고 preemption 기능은 사용하지 않도록 설정하였다. 본 논문의 실험에서는 메인 보드에 장착된 Marvell 사의 GbE 컨트롤러(TG3)와 데이타 송수신 기능을 하드웨어 가속 모듈로 구현한 하이브리드 TOE의 성능을 비교하였다.

그림 5는 일반 GbE 어댑터(TG3)와 하이브리드 TOE의 최소 지연시간을 비교하고 있다. 일반 GbE 어댑터는 최소 지연시간이 약 43 μs로 측정되었으며, 하이브리드 TOE는 약 19 μs의 최소 지연시간을 보였다. 실험 결과 하드웨어로 데이타 패킷을 생성하고 처리하는 하이브리드 TOE가 지연시간 측면에서 우수한 모습을 보였다.

그림 6은 호스트 CPU 상에서 측정된 하이브리드 TOE와 일반 GbE 어댑터의 CPU 점유율을 비교하고 있다. 하이브리드 TOE의 CPU 점유율은 6%를 넘지 않아 일반 GbE 어댑터에 비해 약 10배에 가까운 향상을 보였다.

그림 7은 일반 GbE 어댑터와 하이브리드 TOE의 통신 대역폭을 비교하고 있다. 이 실험에서 하이브리드 TOE는 최대 약 675Mbps의 대역폭을 보였다. 이 결과는 하이브리드 TOE 구조에서 1 Gbps의 속도로 TCP/IP를 처리하기에는 부족한 성능을 가지는 범용 임베디드 프로세서를 사용하더라도 데이타 송수신을 가속시키는 하드웨어의 보조와 소프트웨어/하드웨어 연동 메커니즘을 통해 높은 TCP/IP 처리 성능을 가질 수 있음을 보여준다. 특히 16 KB 미만의 데이타를 전송하는 경우 하이브리드 TOE의 단방향 대역폭은 일반 GbE 어댑터의 성능을 능가하였다.

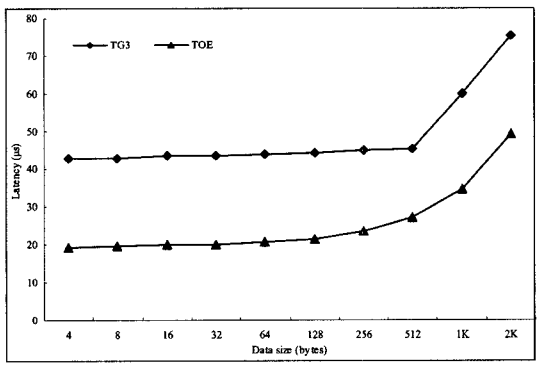

그림 5 일반 GbE 어댑터와 하이브리드 TOE의 지연시간 비교

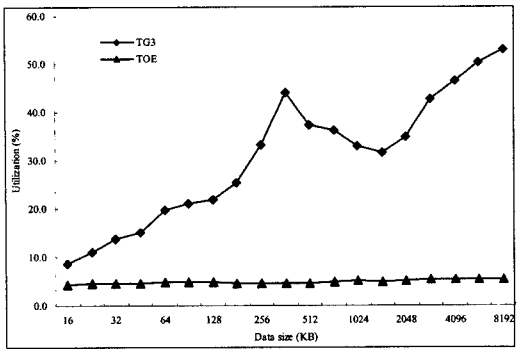

그림 6 일반 GbE 어댑터와 하이브리드 TOE의 CPU 점유율 비교

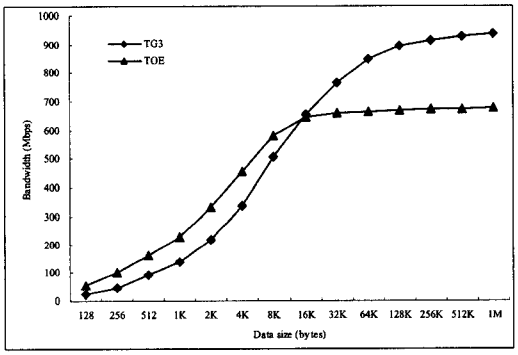

그림 7 일반 GbE 어댑터와 하이브리드 TOE의 대역폭 비교

5. 결론 및 향후 과제

본 논문에서는 하이브리드 TOE에서 통신 성능의 향상을 도모하기 위해 하드웨어 기반의 송수신 가속기를 설계하였다. 데이타 송수신에 필요한 주요 기능들을 하드웨어로 구현하였고, 연결 설정과 흐름 제어 등의 기능은 소프트웨어로 처리하도록 하였다. 그리고 이를 두 개의 프로세서 코어를 내장한 FPGA 상에서 구현하여 하이브리드 TOE의 동작과 성능을 실험하였다. 실험에서는 일반 GbE 어댑터와 하드웨어 가속기를 포함한 하이브리드 TOE의 성능을 비교하여 하드웨어 구현의 성능을 확인하였다. 실험 결과 하이브리드 TOE의 최소 지연시간은 약 19 μs이었으며, 일반 GbE 어댑터에 비해 10배 정도 낮은 수치인 6% 이하의 CPU 점유율로 약 675 Mbps의 최대 대역폭을 제공하였다.

본 연구의 향후 과제로는 하드웨어로 구현된 가속 모듈의 성능을 개선하여 하이브리드 TOE의 성능을 최대화하고자 한다. 그리고 하이브리드 TOE를 기반으로 RDMA 프로토콜이나 iSCSI 프로토콜 등을 효율적으로 오프로딩하는 구조를 개발할 계획이다.

참 고 문 헌

[1] N. Bierbaum, "MPI and Embedded TCP/IP Gigabit Ethernet Cluster Computing," Proceedings of 27th Annual IEEE Conference on Local Computer Networks 2002, pp. 733-734, Nov. 2002.

[2] E. Yeh, H. Chao, V. Mannem, J. Gervais and B. Booth, "Introduction to TCP/IP Offload Engine (TOE)," 10 Gigabit Ethernet Alliance, April 2002.

[3] Intel Corporation, "Intel PRO/1000T IP Storage Adapter," Data Sheet, 2003, available at http://www.intel.com

[4] Boon S. Ang, "An Evaluation of an Attempt at Offloading TCP/IP Protocol Processing onto an i960RN-based iNIC," Technical Reports HPL-2001-8, available at http://www.hpl.hp.com/techreports/2001/HPL-2001-8.html

[5] Alacritech, Inc., "SLIC Technology Overview," Technical Review, 2002, available at http://www.alacritech.com

[6] Available at http://support.qlogic.com

[7] Available at http://www.broadcom.com

[8] Chelsio Communications, "The Unified Wire Engine: Introducing Terminator 3," White Paper, available at http://www.chelsio.com/solutions/pdf/T3_Unified_Wire_Eng_WP.pdf

[9] Available at http://www.neteffect.com

[10] 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, pp. 406-409, 2004.

[11] H. Jang, S.-H. Chung, S.-C. Oh, "Implementation of a Hybrid TCP/IP Offload Engine Prototype," Proceedings of the 10th Asia-Pacific Computer Systems Architecture Conference, pp. 464-477, 2005.

[12] Zhong-Zhen Wu, Han-Chiang Chen, "Design and Implementation of TCP/IP Offload Engine System over Gigabit Ethernet," Proceedings of the 15th International Conference on Computer Communications and Networks, pp. 245-250, Oct. 2006.

[13] S. Aiken, D. Grunwald, A. R. Pleszkun, and J. Willeke, "A Performance Analysis of the iSCSI Protocol," Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies, pp. 123-134, 2003.

[14] Lionbridge Technologies, Inc., "Alacritech SES-1001T: iSCSI HBA Competitive Analysis," VeriTest Benchmark Report, 2004, available at http:// www.veritest.com

[15] H. Ghadia, "Benefits of full TCP/IP offload (TOE) for NFS Services," Proceedings of 2003 NFS Industry Conference, 2003, available at http://nfsconf.com

[16] W. Feng, P. Balaji, C. Baron, L. N. Bhuyan, D. K. Panda, "Performance characterization of a 10-Gigabit

Ethernet TOE," Proceedings of 13th Symposium on High Performance Interconnects, pp. 58-63, Aug. 2005.

장 한 국

1999년 부산대학교 컴퓨터공학과 학사 2001년 부산대학교 컴퓨터공학과 석사 수료. 2001년~현재 부산대학교 컴퓨터공학과 석박사 통합과정. 관심분야는 컴퓨터 구조, 클러스터 시스템, TOE, RDMA

정 상 화

1985년 서울대학교 전기공학과 학사. 1988년 Iowa State Univ. 컴퓨터공학과 석사 1993년 Univ. of Southern California 컴퓨터공학과 박사. 1993년~1994년 Univ. of Central Florida 컴퓨터공학과 조교수. 1994년~현재 부산대학교 컴퓨터공학과 교수, 컴퓨터및정보통신연구소 연구원. 2002년~2003년 Oregon State Univ. 컴퓨터공학과 초빙교수. 관심분야는 클러스터 시스템, 병렬처리, TOE, RDMA, RFID

유 대 현

2007년 부산대학교 컴퓨터공학과 학사 관심분야는 컴퓨터 구조, 클러스터 시스템, TOE, RDMA, 2007년~현재 부산대학교 컴퓨터공학과 석사과정