

# NetFPGA 플랫폼 소개

최훈규 · 권태경 (서울대학교)

## I. 서론

미래 인터넷은 현재의 인터넷이 가진 문제점을 해결하고 미래의 요구사항을 만족시키기 위해 새로운 네트워크의 구조나 프로토콜을 설계하는 연구 분야다. 미래 인터넷 연구를 통해 앞으로 인터넷은 점점 진화해 나갈 것으로 예상된다. 새로운 구조를 설계하고 개발하기 위해 연구자들은 많은 기법들을 제안하고 실험하게 되는데 정확한 실험을 위해서는 자신이 설계한 미래 인터넷 환경을 구축하는 것이 필요하다. 하지만 이미 널리 사용되고 있는 네트워크 장비의 하드웨어나 소프트웨어를 바꿀 수는 없기 때문에 연구실과 같은 작은 규모에서 이런 환경을 구축하는 것은 거의 불가능하다.

NetFPGA<sup>1)</sup>는 학생 또는 연구자가 자신이 설계한 시스템을 하드웨어를 이용하여 구축할 수 있도록 해주는 플랫폼이다. FPGA (Field Programmable Gate Array)가 포함된 NetFPGA 카드를 표준 PC에 장착하면 하나의 NetFPGA 시스템이 완성되기 때문에 저렴한 가격으로 실험 환경을 구축할 수 있다. 또한 공개 소스를 기초로 하여 새로운 설계를 Verilog 코드로 작성하여

FPGA에 로딩함으로써 쉽고 빠르게 새로운 시스템의 프로토타입을 만들어 낼 수 있다. NetFPGA는 하드웨어 재사용 기능<sup>2)</sup>을 제공하기 때문에 미래 인터넷을 연구하는 사람들에게 매우 좋은 도구가 될 것으로 기대된다.

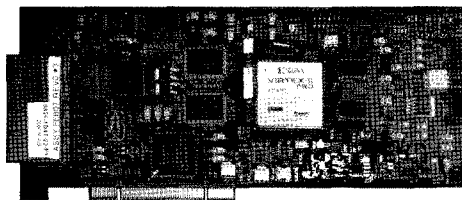
현재 NetFPGA 플랫폼은 전 세계 100개 이상의 그룹에서 이용되고 있으며, NetFPGA를 가지고 연구나 개발을 하고자 하는 사람들에게 Tutorial을 개최하여 NetFPGA를 접할 수 있는 기회를 제공하고 이용을 장려하고 있다. 지금까지 많은 대학과 기관에서 Tutorial 행사가 열려왔으며 본 고에서는 2009년 2월 25~26일, 2일에 걸쳐 서울대학교에서 개최되었던 NetFPGA Tutorial 행사<sup>3)</sup>의 내용을 중심으로 하여 NetFPGA 플랫폼을 소개하고자 한다.

## II. NetFPGA 하드웨어

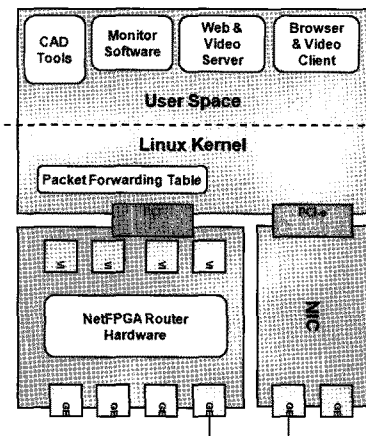
NetFPGA 플랫폼은 네트워킹 하드웨어와 라우터를 디자인하는 도구로써 Stanford 대학에서 처음 개발되었다. NetFPGA는 연구자들이 새로운 아이디어를 고속의 네트워킹 하드웨어 위에서

실험하는 목적으로도 매우 유용하게 이용될 수 있다는 것이 여러 전례를 통해 확인되었다. NetFPGA는 하드웨어의 재사용가능한 특징과 공개 코드에 기반한 특징으로 인해 앞으로 점점 더 많은 보급이 이루어질 것으로 기대된다.

<그림 1>은 NetFPGA 하드웨어의 모습이다. NetFPGA는 User-defined Logic으로 이용되는 Xilinx Virtex-2 Pro FPGA와 PCI Host 인터페이스에 이용되는 Xilinx Spartan FPGA, 2\*2.25MB ZBT SRAM, 64MB DDR2 RAM 등의 메모리, 그리고 4개의 기가비트 이더넷 포트가 이루어져 있다. Standard PC의 PCI 슬롯에 NetFPGA 카드를 장착하면 하나의 NetFPGA 시스템이 완성되며 사용자는 자신의 커스텀 PC를 사용하거나 완성된 NetFPGA 시스템인 Pre-built Cube를 구입하여 사용할 수 있다. NetFPGA 시스템의 일반적인 구성 요소는 <그림 2>와 같다.



<그림 1> Net FPGA Card

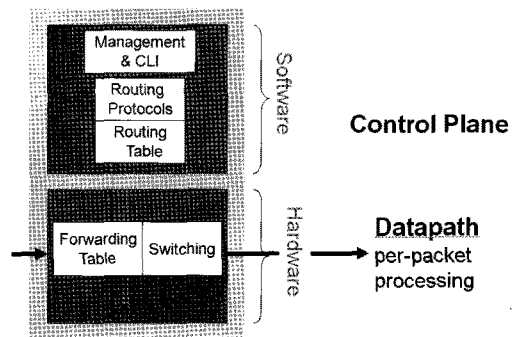


<그림 2> NetFPGA 시스템 구성요소

- 네트워크 : 듀얼 기가비트 이더넷 포트 (Host PCI-E NIC), 4개의 기가비트 이더넷 포트 (NetFPGA)
- 메인보드 : PCI와 PCI-E 슬롯이 장착된 표준 AMD 또는 인텔 x86 컴퓨터
- 프로세서 : 듀얼 또는 쿼드코어 CPU
- 운영체제 : Linux CentOS 5.2

### III. NetFPGA의 이용 모델

NetFPGA 시스템은 하나의 라우터로써 동작한다. <그림 3>에서 볼 수 있듯이 라우팅 프로토콜과 라우팅 테이블은 소프트웨어가, 포워딩 테이블과 스위칭은 하드웨어가 담당하게 된다. NetFPGA 라우터의 장점은 값이 싸며 모든 부분이 완전히 프로그래밍 가능하다는 것이다. 소프트웨어 부분의 C/C++ 에, 하드웨어 부분은 Verilog 에 기반하고 있는데 둘 모두 공개 소스로서 레퍼런스 코드가 제공된다.



<그림 3> IP Router on NetFPGA

공식 홈페이지<sup>1)</sup>에서 얻을 수 있는 NetFPGA Package (NFP) 안에는 네트워킹에 필요한 라우터의 기능을 구현한 소스 코드가 포함되어 있다. 이

코드를 기초로 하여 개발자는 자신의 시스템을 설계할 수 있다. 이 때 개발자가 택할 수 있는 NetFPGA 이용 모델은 크게 세 가지로 나뉘볼 수 있다.

### 1. 소프트웨어의 수정

레퍼런스 라우터 하드웨어가 FPGA 안에 로딩된 상태에서 소프트웨어를 수정하여 새로운 커스텀 프로토콜 구현하는 방법이다. 이 경우에 NetFPGA 보드는 IPv4 하드웨어로 프로그래밍되어 있으며 Linux 호스트는 NFP에서 배포되는 라우터 키트를 사용한다. 라우터 키트 데몬은 라우팅 테이블과 ARP 캐시를 소프트웨어로부터 하드웨어에 있는 테이블로 미러링하며 이것은 IPv4 라우팅을 가능하게 해 준다.

### 2. 하드웨어의 수정

레퍼런스로 제공된 하드웨어 코드에서 출발, NFP의 라이브러리에 있는 모듈 또는 자신이 작성한 Verilog 코드를 이용하여 수정하는 것이다. 소스 코드를 컴파일 할 때는 표준 디자인 도구가 사용된다. 이렇게 만들어진 Bitfile은 NetFPGA 보드로 로딩된다. 새로운 라우터의 기능 구현은 추가적인 소프트웨어를 구현하거나 기존 소프트웨어의 추가를 통해 보완될 수 있다. 예를 들어, IPv4에서 현재 구현되어 있는 CAM LPM (Longest Prefix Matching) 알고리즘 대신에 Trie LPM을 구현하여 대체하는 것이 가능하다.

### 3. 완전히 새로운 설계

사용자는 자신만의 로직과 데이터 처리 기능

을 직접 FPGA에 구현하는 것으로 완전히 새로운 설계를 할 필요가 있는 경우에 해당한다. 설계를 위해 NFP에서 제공하는 라이브러리를 이용하거나 제 3자의 모듈을 이용할 수 있으며 아예 완전히 새로운 소스 코드를 작성할 수도 있다.

## IV. 튜토리얼 데모

2009년 2월 25-26일, 2일동안 서울대학교에서 Tutorial 행사<sup>3)</sup>가 열렸으며, Cambridge 대학의 Andrew W. Moore 교수와 Brno 대학의 박사과정 학생이자 하드웨어 개발자인 Martin Zadnik가 강사로 참여하였다. 이 행사에서는 NetFPGA를 연구 혹은 개발에 이용하고자 하는 사람들이 쉽게 접근할 수 있도록, 그리고 앞으로의 적극적인 이용을 장려하기 위해 NetFPGA 전반적인 내용 설명과 함께 간단한 예제 등을 시연하였다. 이 장에서는 Tutorial의 시연 내용을 요약하여 소개한다.

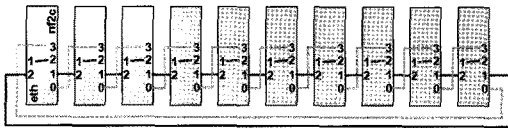
### 1. 토폴로지 구성

NetFPGA 시스템이 갖춰진 10대의 머신을 하나

<표 1> 각 인터페이스 역할

NetFPGA 기가비트 이더넷 인터페이스	
nf2c3	좌측 인접 머신과 연결
nf2c2	로컬 호스트 인터페이스
nf2c1	인접 서버를 위한 루트
nf2c0	우측 인접 머신과 연결
호스트 이더넷 인터페이스	
eth1	로컬 호스트 인터페이스
eth2	인접 머신의 서버

의 클러스터로 묶어 케이블을 연결하고 링(Ring) 형태의 토폴로지를 구성하였다. 앞서 하드웨어 구조에서 언급했듯이 NetFPGA 시스템에는 4개의 기가비트 NetFPGA 이더넷 인터페이스(nf2c0-nf2c3)와 2개의 호스트 이더넷 인터페이스(eth1-2)가 존재한다. 각 인터페이스는 Category 6의 이더넷 케이블로 연결된다. 표 1과 그림 4는 각 인터페이스 역할과 그에 따른 토폴로지를 보여준다. NetFPGA의 포트2는 로컬에서의 클라이언트/서버를 재현하기 위해 이용되며, 포트0과 포트3은 다른 머신들과의 연결에 이용된다. 포트1은 인접 머신과의 서버 기능을 위해 연결되었다.



〈그림 4〉 토폴로지 구성

## 2. 라우팅 테이블 관찰

앞에서 구성한 토폴로지 상에서 비디오 스트리밍을 하고 라우팅 테이블을 관찰하였다. 각 머신은 모두 비디오 서버 혹은 클라이언트가 될 수 있다. 모든 머신은 서버로부터 동영상을 재생한다. 비디오 클라이언트로 Linux mplayer를 사용하였으며 MPEG2 방식으로 비디오 트래픽을 전송하였다. NPF에서 제공되는 자바 그래픽 모듈을 실행하면 라우팅 테이블의 현재 상태를 GUI(Graphic User Interface)로 관찰할 수 있다. 레퍼런스로 제공되는 라우팅 프로토콜은 PW-OSPF이며, 라우팅 테이블의 내용은 목적지의 주소와 그곳까지 최단 거리로 도달하기 위해 선택해야 할

다음 인터페이스에 대한 정보(Next-hop)가 들어 있다. 네트워크가 링(Ring)의 형태를 취하고 있기 때문에 한 머신은 좌우 두 개의 머신과 연결되어 있다. 따라서 한 머신으로부터 각 머신까지의 경로는 좌측 머신을 향한 인터페이스, 우측 머신을 향한 인터페이스로 절반씩 나누어진다.

Index	Destination IP	Subnet Mask	NextHop IP
0	192.168.30.0	255.255....	192.168.2...
1	192.168.29.0	255.255....	192.168.2...
2	192.168.28.0	255.255....	192.168.2...
3	192.168.27.0	255.255....	0.0.0.0
4	192.168.26.0	255.255....	0.0.0.0
5	192.168.25.0	255.255....	0.0.0.0
6	192.168.24.0	255.255....	0.0.0.0
7	192.168.23.0	255.255....	192.168.2...
8	192.168.22.0	255.255....	192.168.2...
9	192.168.21.0	255.255....	192.168.2...
10	192.168.20.0	255.255....	192.168.2...

〈그림 5〉 라우팅 테이블의 예

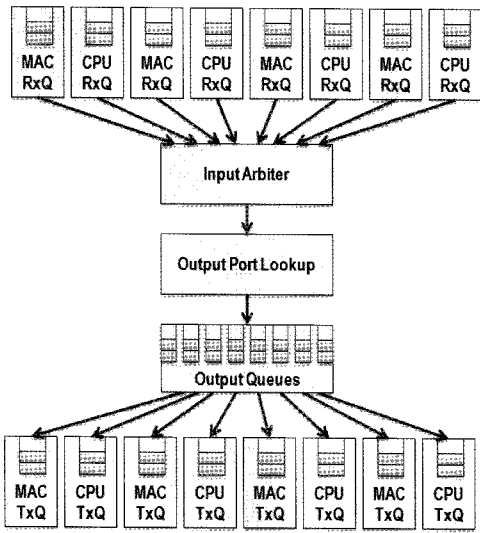
머신 간에 연결된 케이블 하나를 제거하여 임의의 링크를 끊어보면, 라우팅 테이블에서 일부 머신으로의 Next-hop이 변화하는 것을 관찰할 수 있다. 라우팅 프로토콜이 자동으로 새로운 경로를 찾기 때문이다. 이 경우에 재생중인 동영상은 순간적으로 정지되지만 곧바로 다시 재생된다.

## 3. 레퍼런스 라우터 다운로드

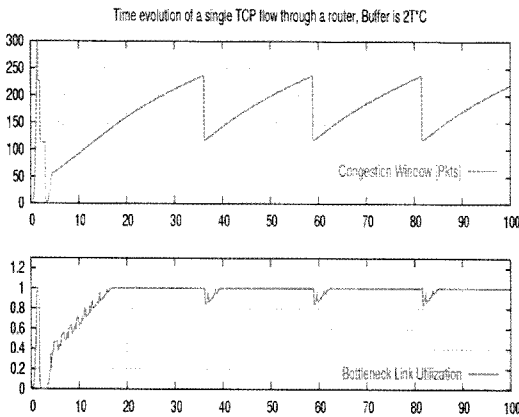
레퍼런스 라우터의 구조는 그림 6과 같다. 각 인터페이스로 들어온 패킷들은 Queue에서 대기하다가 차례로 Input Arbiter로 들어간다. Arbiter는 들어온 패킷들을 스케줄링하여 적절한 순서로 내보낸다. Lookup 과정을 통해 나가야 할 포트가 정해지면 각 패킷은 Output Queue로부터 각 인

터페이스로 빠져나간다.

레퍼런스 라우터에 대한 모든 소스코드는 NFP 안에 포함되어 있다. 레퍼런스 라우터는 해당 디렉토리에서 make 명령을 수행하면 NetFPGA 보드로 다운로드된다. 만일 레퍼런스 라우터의 코드를 수정하였다면 NetFPGA 보드에 다운로드하기 전에 코드를 합성하는 추가적인 과정이 필요하다. 이에 대해서는 다시 언급할 것이다.



〈그림 6〉 레퍼런스 라우터



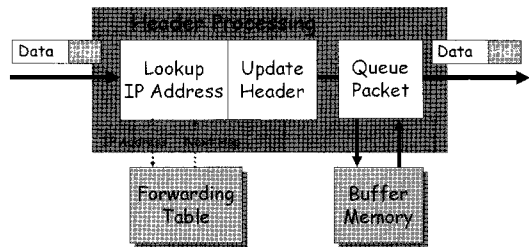
〈그림 7〉 TCP flow 변화

#### 4. TCP Flow 변화 관찰

NFP의 GUI 모듈은 라우팅 테이블 뿐 아니라 현재 Queue의 점유율, TCP Flow, 링크 이용률, 전송된 데이터의 양과 같은 전반적인 정보를 모두 보여준다. iperf 스크립트를 실행하여 현재의 TCP flow를 그래프로 볼 수 있다. 또한 Event Capture나 Rate limiter 등의 모듈을 추가하면 GUI 상에서 이들을 실시간으로 조작하는 것도 가능하다. 그림 7은 TCP flow와 링크 이용률의 예이다.

#### 5. 라우터의 기능 강화

라우터의 일반적인 Datapath 구조는 그림 8과 같다. 들어온 패킷은 대해 헤더 프로세싱을 거쳐 Output Port로 나가게 된다. Datapath 코드에서 Delay를 조절하여 라우터의 기능을 강화할 수 있다. Verilog 소스 코드의 확장자는 .v 파일로 되어 있으며 Emacs와 같은 텍스트 편집기를 이용하여 수정가능하다. 주의할 점은 Verilog 코드를 수정하고 이를 반영하기 위해서는 합성(Synthesis) 과정을 반드시 거쳐야 한다. 합성은 Verilog 코드로부터 하나의 하나의 Bitfile을 생성하는 과정을 말하는데 레퍼런스 라우터의 합성에는 Xilinx의 TEMAC core license<sup>[5]</sup>가 필요하며 생성에 약 한



〈그림 8〉 라우터의 일반적인 Datapath 구조

시간 정도의 시간이 소요된다. 이 Bitfile을 NetFPGA 보드로 다운로드해야 비로소 수정한 부분이 NetFPGA에 반영된다.

## 6. N번 째 패킷 드랍

마지막 시연은 라우터에 N번째로 들어온 패킷을 드랍하도록 수정해보는 것이다. Verilog 코드에 카운터를 추가해 봄으로써 하드웨어 언어가 가진 고유의 특징을 이해할 수 있다. 하드웨어 언어인 Verilog는 C와 같은 구조적 프로그래밍 언어와는 근본적인 차이점이 있다. C 프로그래밍에서는 하나의 로직이 하나의 흐름을 가지고 순차적으로 수행되는 반면 Verilog에서는 로직이 회로의 한쪽 끝에서 반대 쪽 끝을 향해 퍼져나가면서 각 Statement가 동시에 수행된다. 따라서 Verilog 프로그래밍에서 원하는 결과를 얻기 위해서는 이러한 병렬성을 주의깊게 고려하고 동시에 오퍼레이션의 모호함을 최대한 배제하는 것이 필수적이다. 수정된 라우터를 완성하고 성능을 측정해 보는 것으로 Tutorial의 시연은 마무리되었다.

## V. NetFPGA 이용 현황

지금까지 소개한대로 NetFPGA 플랫폼은 저렴한 가격과 공개 코드의 제공, 하드웨어 재사용을 통해 빠른 프로토타입 구축을 가능하게 해 준다. 현재 전 세계 15개 이상의 나라의 120여 기관에 약 1,000여 개의 NetFPGA 시스템이 보급되었으며 점점 더 많은 학교와 기관에서 NetFPGA를 수업 또는 연구를 목적으로 이용하는 추세다. NetFPGA가 처음 개발된 Stanford 대학을 시작으로 Rice 대학과 Cambridge 대학 등지에서는 이미

프로젝트 과목을 통해 2-3명의 학생들이 한 팀이 되어 실제로 새로운 시스템을 설계, 구현하고 테스트하는 기회를 제공하고 있다. 표 2는 NetFPGA를 이용한 수업의 예이다.

〈표 2〉 NetFPGA를 이용한 수업

<b>Stanford University</b>
EE109, "Build an Ethernet Switch"
CS344 "Building an Internet Router"
<b>Rice University</b>
"Network Systems Architecture"
<b>Cambridge University</b>
"Build an Internet Router"

〈표 3〉 NetFPGA를 이용한 연구

<b>Managed flow-table switch</b>
<a href="http://OpenFlowSwitch.org">http://OpenFlowSwitch.org</a>
<b>Buffer Sizing</b>
버퍼 사이즈 감소, 버퍼 점유율 측정버
<b>RCP: Congestion Control</b>
새로운 패킷을 해석, 재기록하는 모듈 개발
<b>Deep Packet Inspection (FPX)</b>
TCP/IP Flow 재구축, 정규 표현식 매칭
<b>Packet Monitoring (ICSI)</b>
로드 밸런싱
<b>Precise Time Protocol (PTP)</b>
라우터 사이의 동기화

또한 개발자 커뮤니티를 통해 NetFPGA를 위한 연구 프로젝트도 활발히 이루어지고 있으며 세계 각지에서 Tutorial 행사를 통해 NetFPGA의

보급과 발전에 기여하고 있다. 표 3은 NetFPGA를 이용하여 진행된 연구 프로젝트들이다.

## VI. 향후 전망

앞으로 NetFPGA로 구현해 볼 수 있는 프로젝트 아이디어로는 다음과 같은 것들이 있다.

- IPv6 라우터
- TCP 트래픽 생성기
- 로드 밸런싱
- 그래픽 유저 인터페이스 (CLACK 같은)
- MAC-in-MAC 캡슐화
- 암호/복호와 모듈
- RCP Transport 프로토콜
- Packet 필터링 (Firewall, IDS, IDP)
- TCP Offload 엔진
- DRAM 패킷 큐
- Sata 브릿지를 이용한 8-Port 스위치
- 독자적인 MAC 구축
- Register 정의를 위한 XML 이용

NetFPGA은 학생들과 연구자들로 하여금 소프트웨어가 아닌 하드웨어의 지원을 통한 새로운 시스템 구축을 가능하게 해주는 플랫폼이기 때문에 종전까지의 미래 인터넷을 위한 테스트베드와는 차별되는 장점을 가지고 있다. NetFPGA를 이용하고자 하는 사람은 공식 홈페이지<sup>[1]</sup>를 방문하고 커뮤니티에 가입하여 연구와 개발에 도움을 얻을 수 있으며 많은 사람들의 참여와 기여를 통해 향후 NetFPGA 프로젝트의 지속적인 발전이 기대된다.

## 참고문헌

- [1] <http://www.netfpga.org>
- [2] J.Naous, G.Gibb, S.Bolouki, N.McKeown, SIGCOMM PRESTO Workshop, August 2008.
- [3] <http://fir.kr/netfpga>
- [4] <http://www.xilinx.com>
- [5] V2PRO TEMAC core  
<http://www.xilinx.com/products/ipcenter/TEMAC.htm>

저자소개



최 훈 규

2008년 2월 성균관대학교 정보통신공학부 학사  
 2009년 현재 서울대학교 컴퓨터공학부 석사과정  
 재학중  
 주관심 분야 : Future Internet, Addressing

저자소개



권 태 경

1993년 2월 서울대학교 컴퓨터공학부 학사  
 1995년 2월 서울대학교 컴퓨터공학부 석사  
 1998년 6월 Visiting student, IBM Thomas  
 J. Watson Research Center  
 1999년 1월 Visiting scholar, University of  
 North Texas  
 2000년 2월 서울대학교 컴퓨터공학부 박사  
 2000년 3월 Part-time lecturer, Hanyang  
 University  
 2000년 9월 ~ 2000년 11월 Post-doctoral  
 Researcher, Soongsil  
 University  
 2000년 12월 ~ 2002년 8월 Post-doctoral  
 Researcher, UCLA  
 2002년 9월 ~ 2003년 3월 Principal  
 Engineer, IST International  
 Inc.  
 2003년 3월 ~ 2004년 1월 Post-doctoral  
 Researcher, City University  
 New York  
 2004년 2월 ~ 2008년 3월 Assistant  
 Professor, Seoul National  
 University  
 2008년 4월 Present: Associate Professor,  
 Seoul National University  
 주관심 분야 : wireless networks: ad hoc network,  
 sensor network, multimedia streaming wireless  
 technologies convergence mobile networks:  
 mobility management, peer-to-peer mobility  
 ubiquitous/mobile computing Internet: active  
 queue management, peer-to-peer