

NetFPGA 기반 OpenFlow switch 설치 및 향후 계획

충남대학교 정보통신공학부
이재용





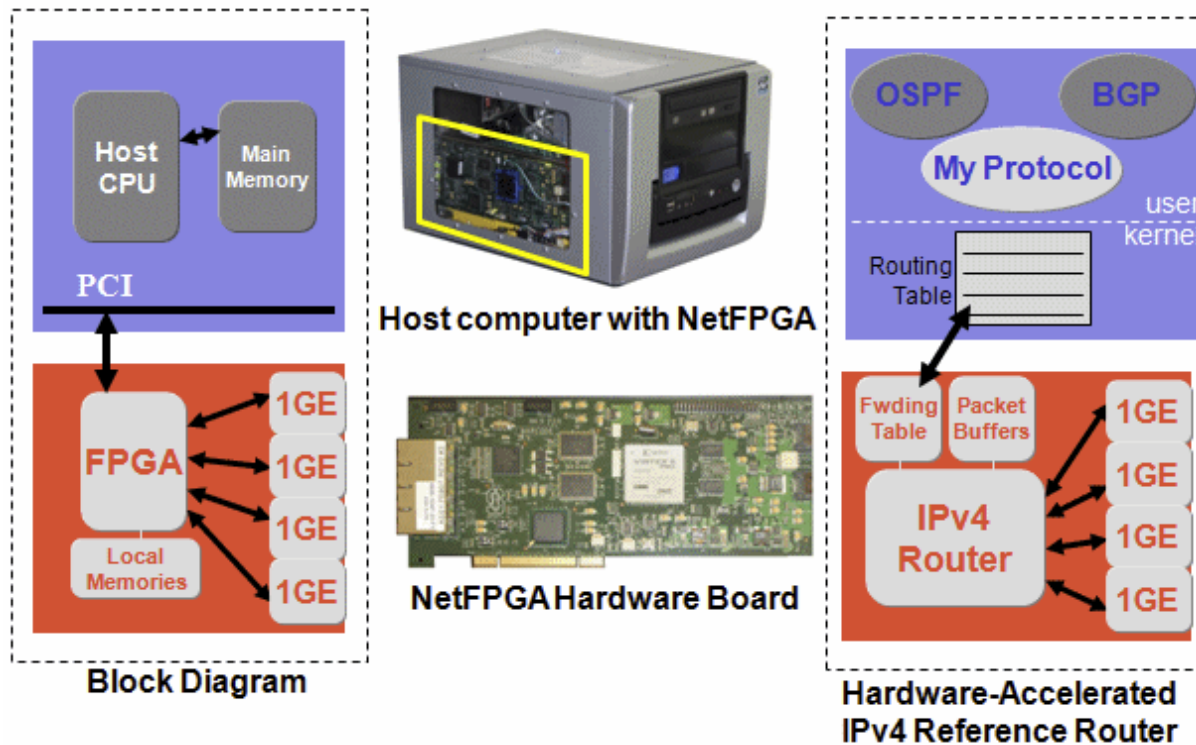
목 차

- ❖ NetFPGA 설치 및 테스트
- ❖ OpenFlow switch 설치 및 테스트
- ❖ 향후 연구



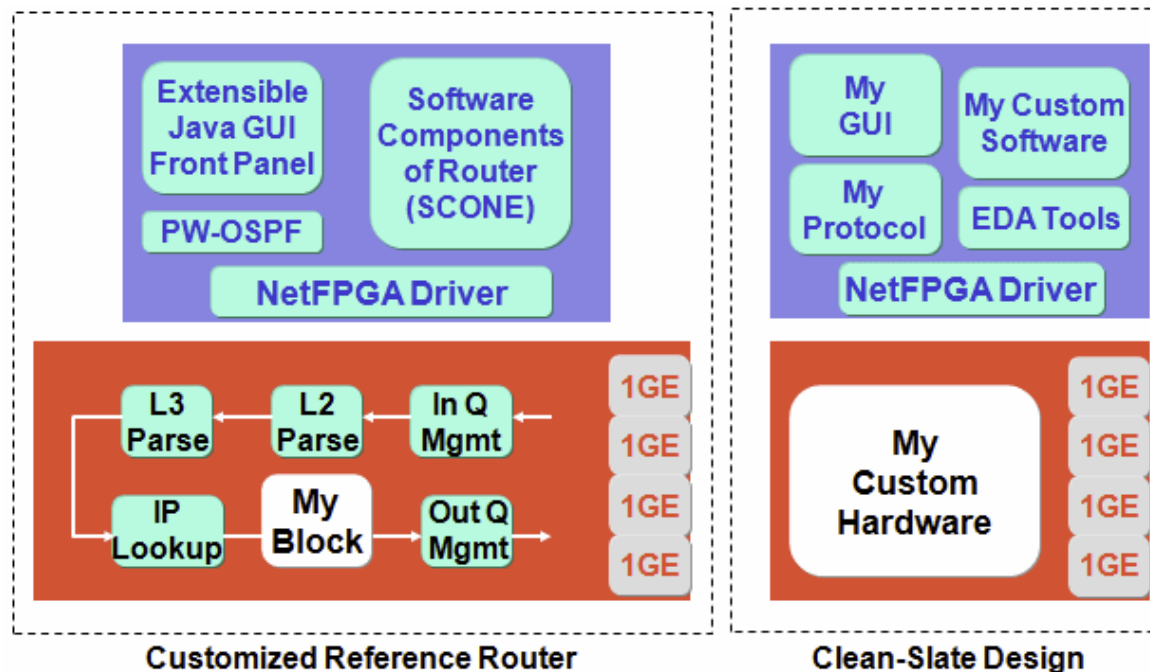
NetFPGA Introduction

- ❖ The NetFPGA is a low-cost platform
 - Tool for teaching networking hardware and router design
 - Tool for networking researchers



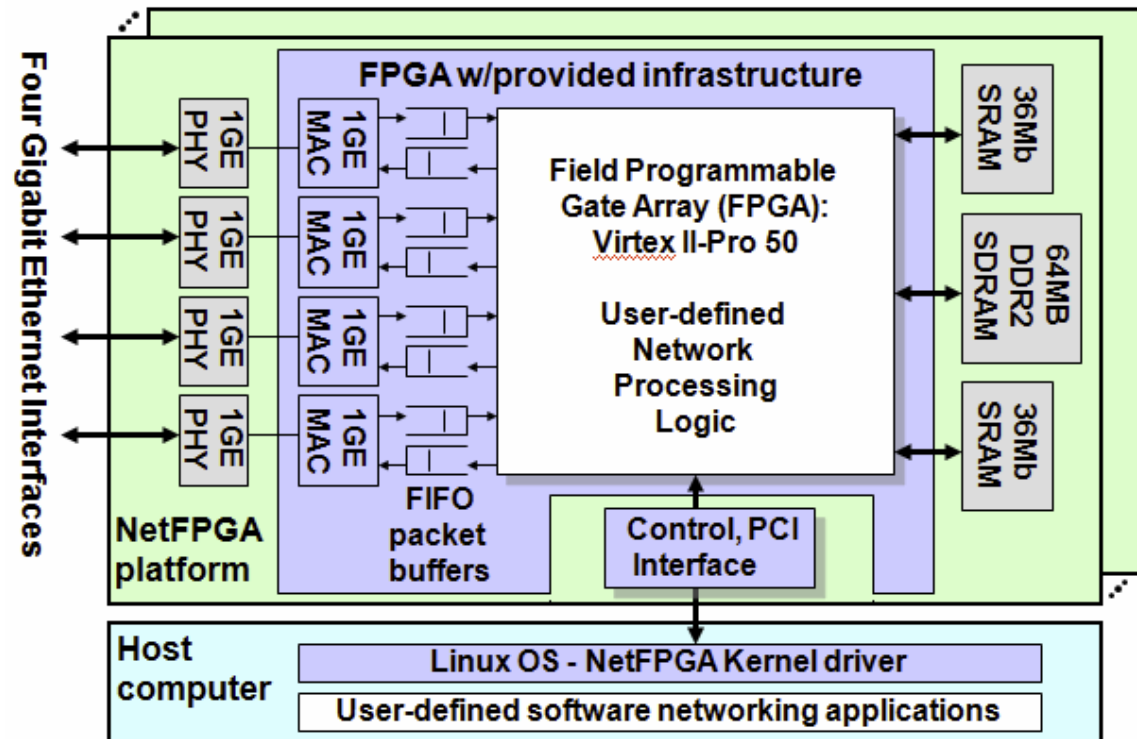
NetFPGA Usage models

- ❖ 1) Reference Router : can be downloaded from <http://NetFPGA.org>
 - Hardware-accelerated NIC, or IPv4 router
 - Openflow switch extension
- ❖ 2) Customized Router
- ❖ 3) Complete new design



NetFPGA Major components

- ❖ Interfaces: PCI, 4GE
- ❖ Memories
- ❖ FPGA resources
- ❖ Schematic and board layout
 - PDF
(http://netfpga.org/NetFPGA_PCB_r3.pdf)
 - Orcad Capture file
(http://netfpga.org/NetFPGA_PCB-May-14-2008.zip)



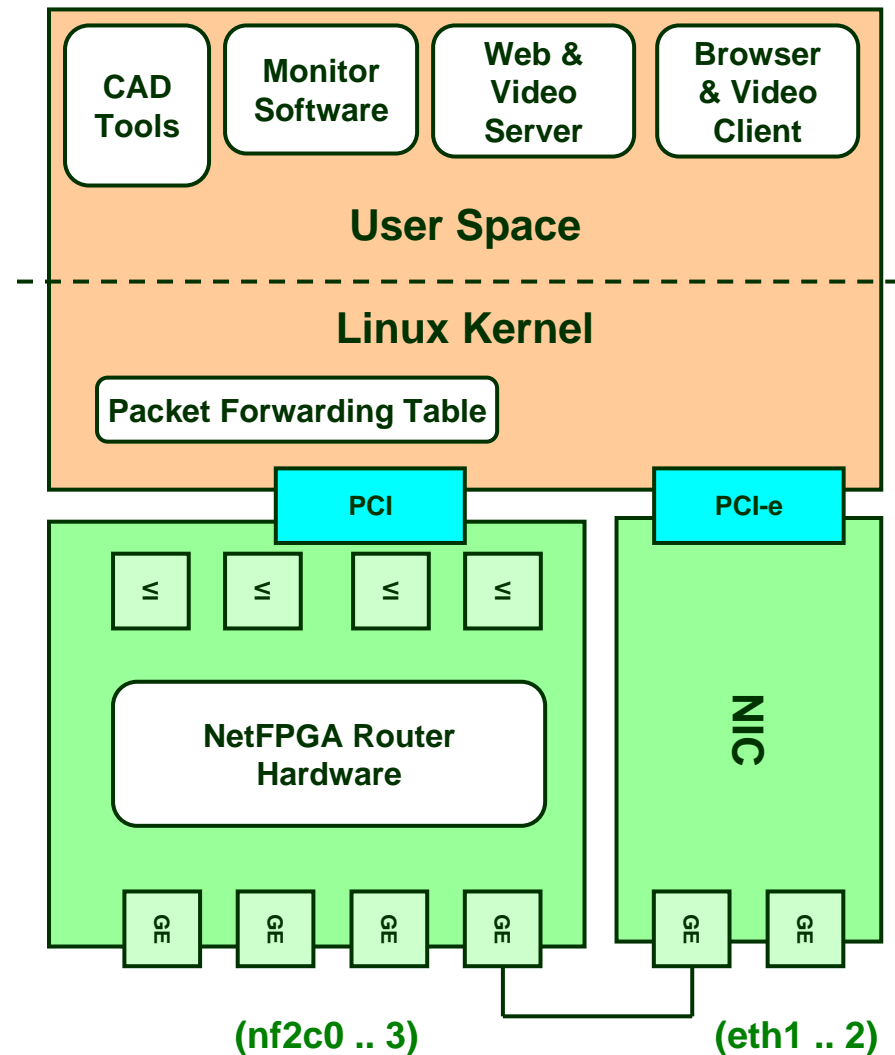
Obtaining NetFPGA system

❖ Obtaining NetFPGA Hardware

- Digilent Inc. ,
sales@digilentinc.com

❖ Obtaining a Host PC for the NetFPGA

- Assemble your own PC using from components
- Purchase a Dell 2950
- Purchase a complete pre-built system





Obtaining Gateway/Software Package

- ❖ Download the Beta NetFPGA Package (NFP)
 - Sign up for a new beta account
 - ❑ <http://netfpga.org/netfpgawiki/index.php?title=Special:Userlogin&type=signup>
 - Download the tarballs from <http://NetFPGA.org/beta/distributions>
 - ❑ [netfpga_base_beta_1_2_5_2_CentOS5.tar.gz](#)
- ❖ Download the extended NetFPGA Package (BetaPlus, optional)
 - 1) Complete the survey on-line as: <http://netfpga.org/survey.html>
 - 2) Send an email to jwlockwd@stanford.edu
 - ❑ written guarantee that you will not re-distribute the source code
 - 3) 7 days for review → email confirmation
 - 4) download the NFP from <http://NetFPGA.org/betaplus>
 - ❑ [netfpga_full_beta_1_2_5_1.tar.gz](#)



Install Software

- ❖ CentOS 설치 (4.4, 4.5, 5.1, 5.2)
- ❖ GUI 환경을 위한 JDK 설치
 - 1.6 이후 버전 설치
- ❖ RPMforge repository
 - yum 이용 패키지 인스톨시 인스톨 정보를 얻을 수 없을 경우 rpmforge를 repository로써 추가시키면 최신정보에 의해 인스톨 가능
- ❖ NetFPGA Base Package
- ❖ Verilog 개발툴 설치
 - Xilinx ISE , ModelSim, ChipScope
- ❖ 시뮬레이션을 위한 Memory Modules 추가
 - Micron DDR2 SDRAM
 - Cypress SRAM



Verify the software and hardware

❖ Compile and Load Driver

- cd ~/NF2/
- make
- make install
- reboot

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# lsmod | grep nf2  
nf2                22156  0  
[root@localhost ~]# ifconfig -a | grep nf2  
nf2c0      Link encap:Ethernet  HWaddr 00:4E:46:32:43:00  
nf2c1      Link encap:Ethernet  HWaddr 00:4E:46:32:43:01  
nf2c2      Link encap:Ethernet  HWaddr 00:4E:46:32:43:02  
nf2c3      Link encap:Ethernet  HWaddr 00:4E:46:32:43:03  
[root@localhost ~]#
```

Verify the software and hardware

❖ Run Selftest

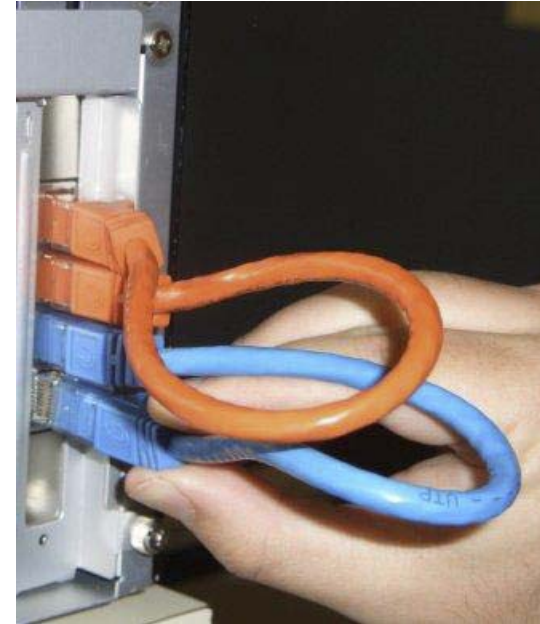
➤ Reprogram the CPCI

- ❑ /usr/local/sbin/cpci_reprogram.pl --all
- ❑ PC를 restart 할 때마다 시행

➤ Loopback cable 설치

➤ Load self-test bitfile

- ❑ nf2_download ~/NF2/bitfiles/selftest.bit



```
[root@localhost ~]# ifconfig nf2c3 up
[root@localhost ~]# ~/NF2/projects/selftest/sw/selftest
Found net device: nf2c0
NetFPGA selftest 1.00 alpha
Running..... PASSED
[root@localhost ~]#
```

Verify the software and hardware

❖ Run Regression Tests

➤ Connect Ethernet test cables

➤ Load reference_router bitfile

❑ nf2_download ~/NF2/bitfiles/reference_router.bit

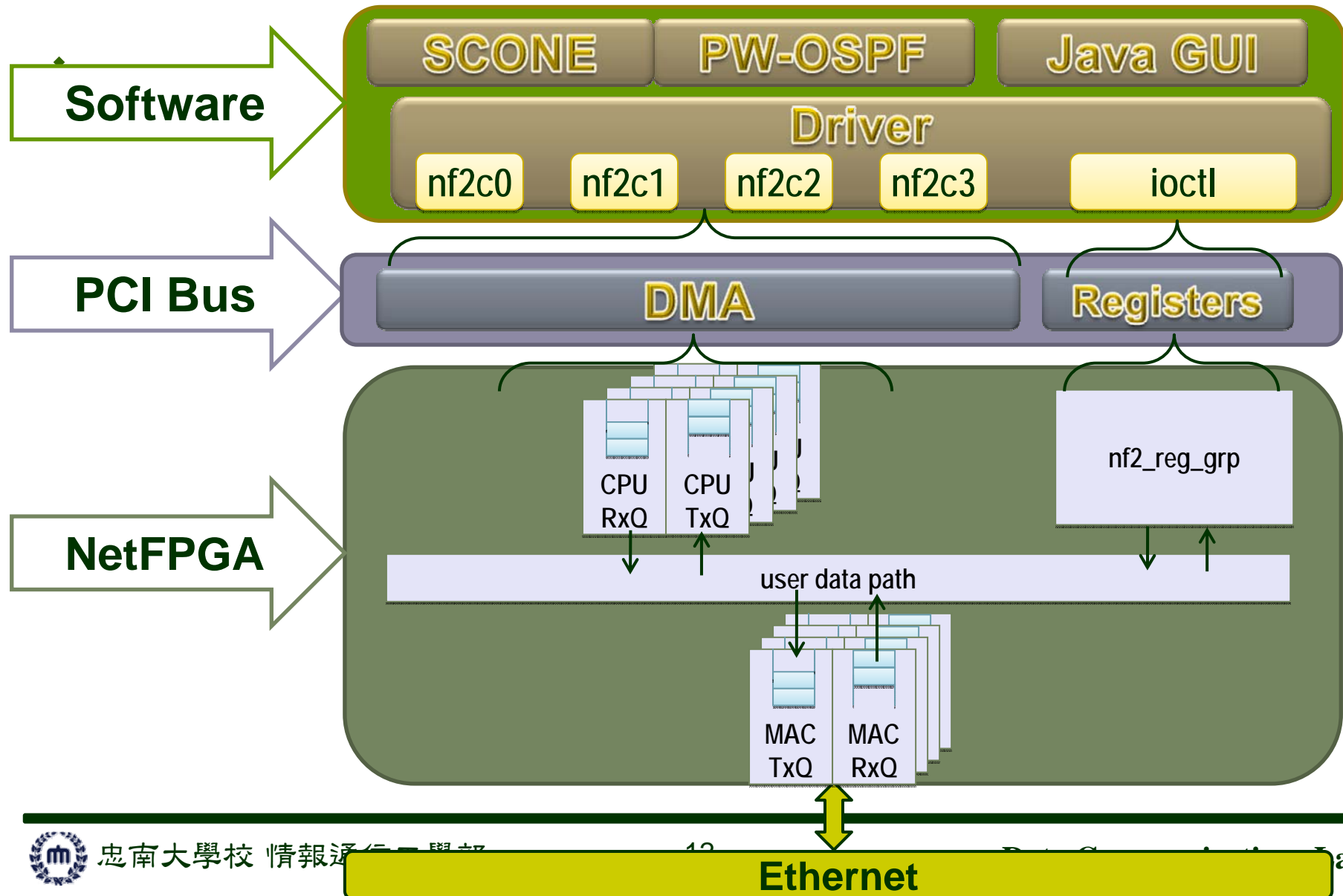
➤ Run regression test suite

❑ ~/NF2/bin/nf21_regress_test.pl

```
1 Root directory is /root/NF2
2 Running tests on project 'driver'...
3   Running global setup... PASS
4   Running test 'driver_compile'... PASS
5   Running test 'driver_install'... PASS
6   Running test 'modprobe'... PASS
7   Running test 'verify_mtu'... PASS
8   Running test 'verify_4_interfaces'... PASS
9   Running global teardown... PASS
10
11
12 Running tests on project 'reference_nic'...
13   Running global setup... PASS
14   Running test 'download_nic'... PASS
15   Running test 'test_loopback_random'... PASS
16   Running test 'test_loopback_minsize'... PASS
17   Running test 'test_loopback_maxsize'... PASS
18   Running test 'test_loopback_drop'... PASS
19   Running test 'test_loopback_crc'... PASS
20   Running test 'test_ip_interface'... PASS
21   Running global teardown... PASS
22
```



NetFPGA full system components





The SCONE

❖ Software Component Of NetFPGA

- IPv4 forwarding,
- handles ARPs and various ICMP messages,
- has telnet (port 23) and web (port 8080) interfaces
- implements a subset of OSPF named PW-OSPF.

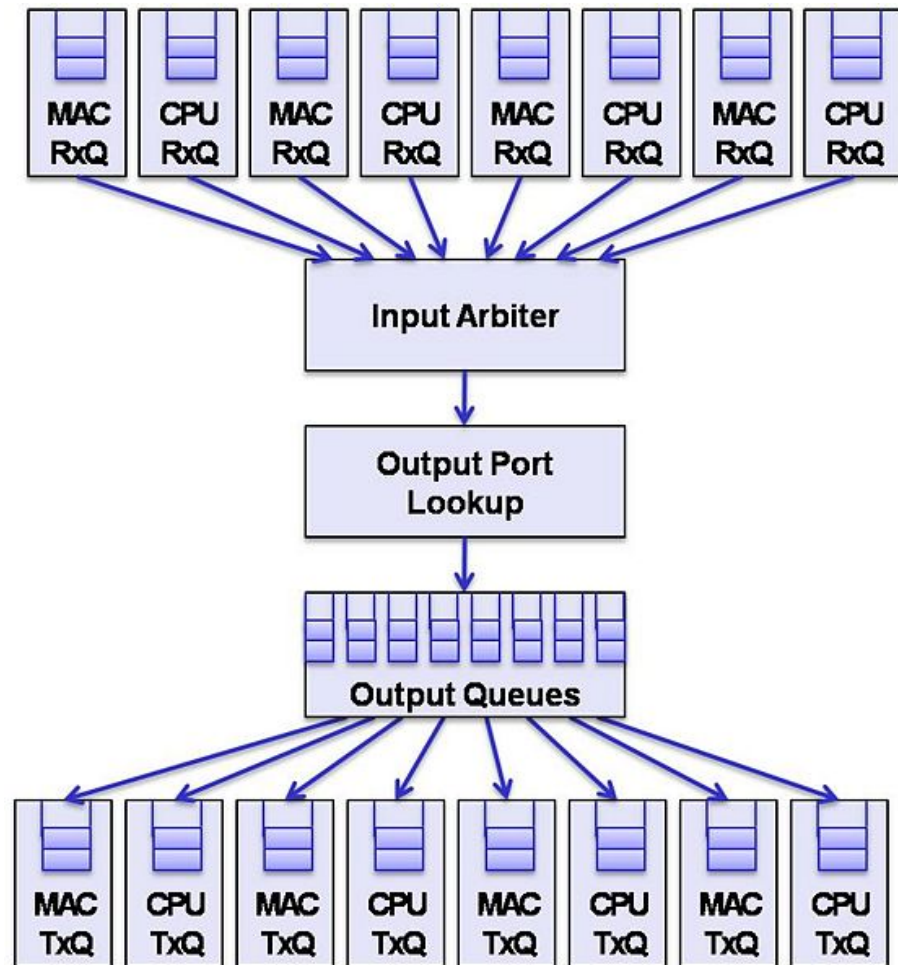
❖ SCONE mirrors

- a copy of its MAC addresses, IP addresses, routing table, and ARP table to the NetFPGA card
- which hardware accelerates the forwarding path



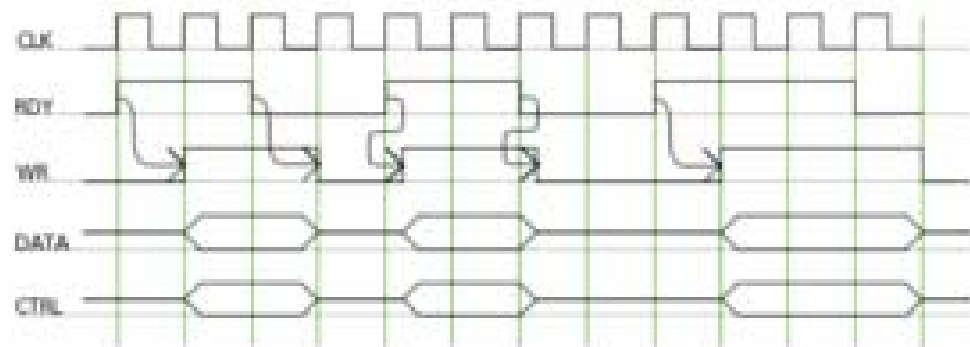
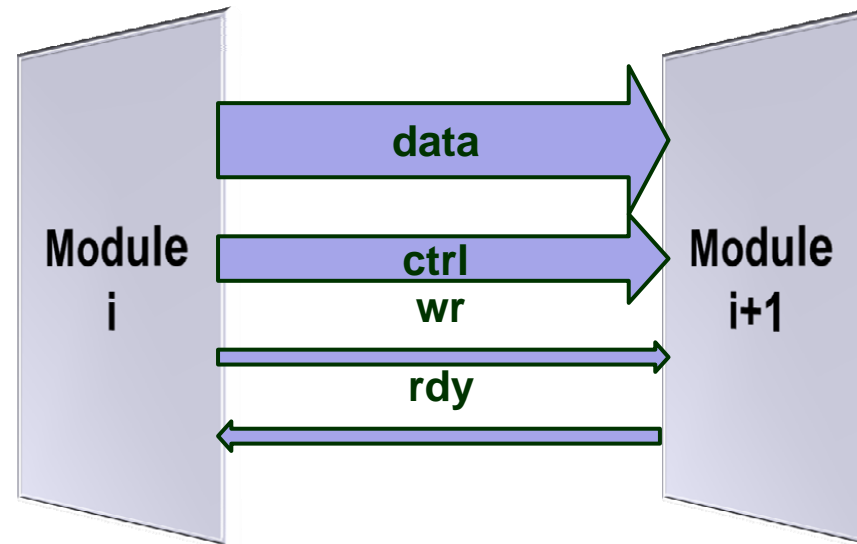
The reference pipeline of NetFPGA

- ❖ Rx queues module
- ❖ Input arbiter module
 - 다음 서비스를 위한 RxQ 결정
- ❖ Output port lookup module
 - 출력 포트 결정
 - “Openflow switch” 확장 module
- ❖ Output queues module
 - 출력포트 packet 일시 저장
 - “Traffic scheduler “ 구현 위치
- ❖ Tx queues module



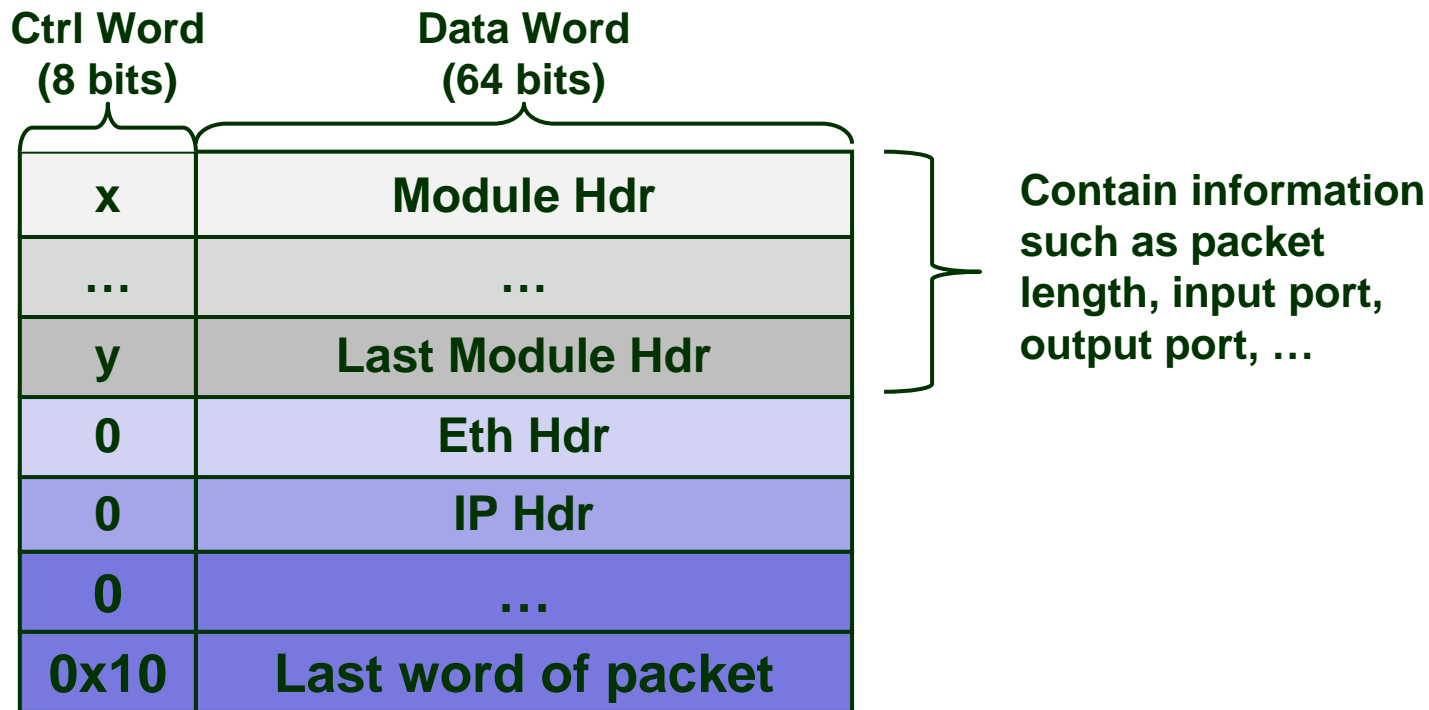
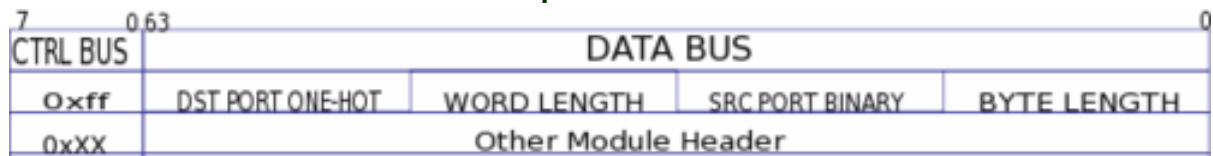
Inter-Module Communication

- ❖ Only “Push” operation
 - “RDY” signal : 1 bit
 - “WR” signal : 1 bit
 - “CTRL” signal: 8 bit
 - Module header 구분
 - End of Packet 구분
 - “DATA” signal : 64 bit



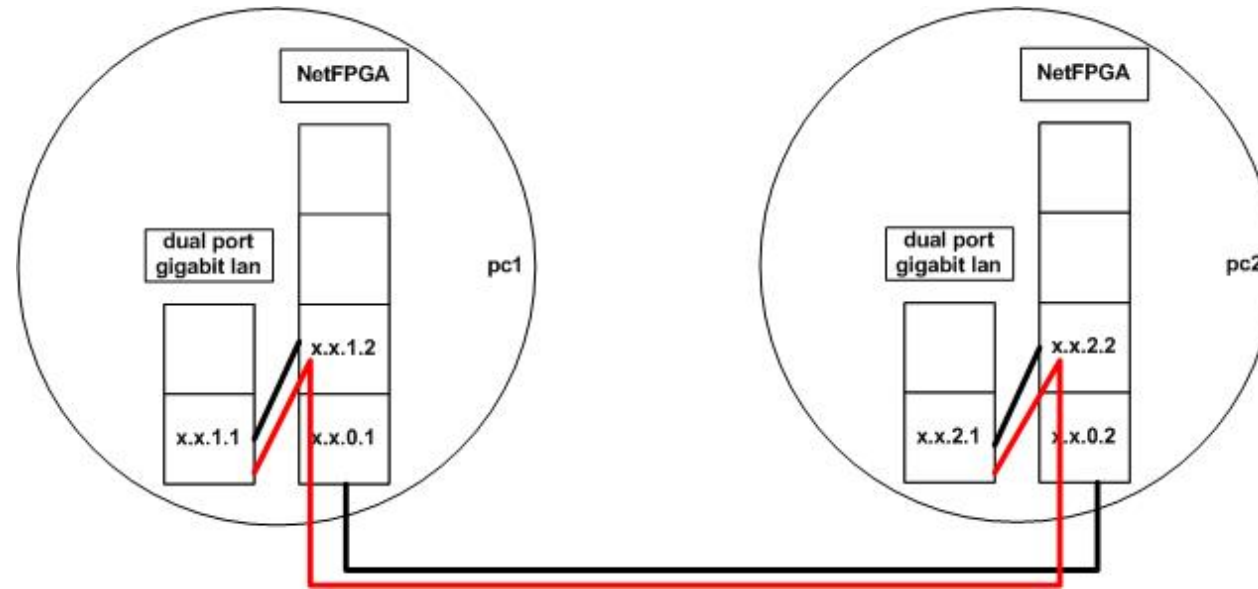
Inter-Module Communication

- ❖ “Module header” 로 정보 전달
 - First module header: Rx queue 에서 생성



NetFPGA 라우팅 테스트

❖ 실험 환경



❖ 실험 결과

- ping 을 이용 확인

```
[root@localhost ~]# ping -I 192.168.1.1 192.168.2.1 -c 10
PING 192.168.2.1 (192.168.2.1) from 192.168.1.1 : 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=0 ttl=62 time=0.140 ms
64 bytes from 192.168.2.1: icmp_seq=1 ttl=62 time=0.121 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=62 time=0.187 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=62 time=0.125 ms
```

NetFPGA 대역폭 측정 실험

❖ 실험환경

➤ iperf 프로그램

➤ Reno TCP

➤ t = 1800s

➤ 단말 환경

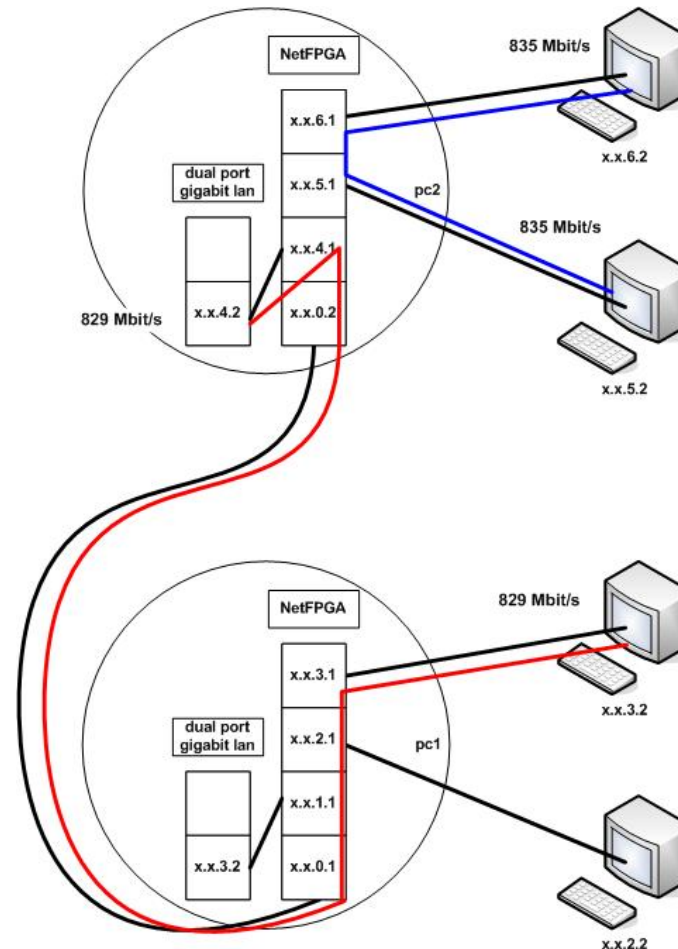
❑ OS : Cent OS 5.2

(Kernel 2.6.18-web100)

❑ CPU : 2.8GHz

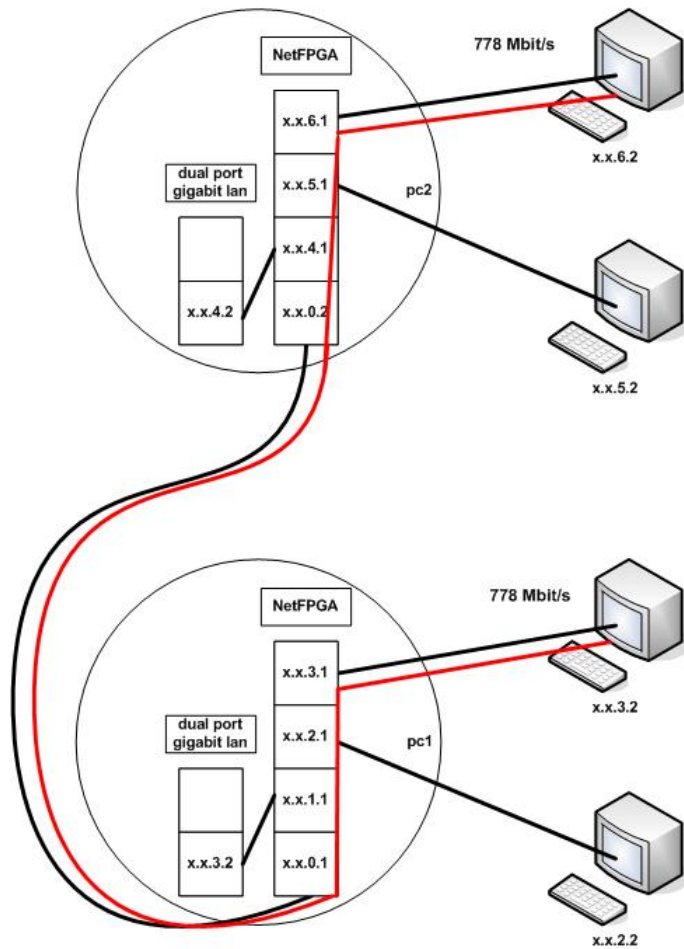
❑ RAM : 512MB

❑ LAN : dual port 1G

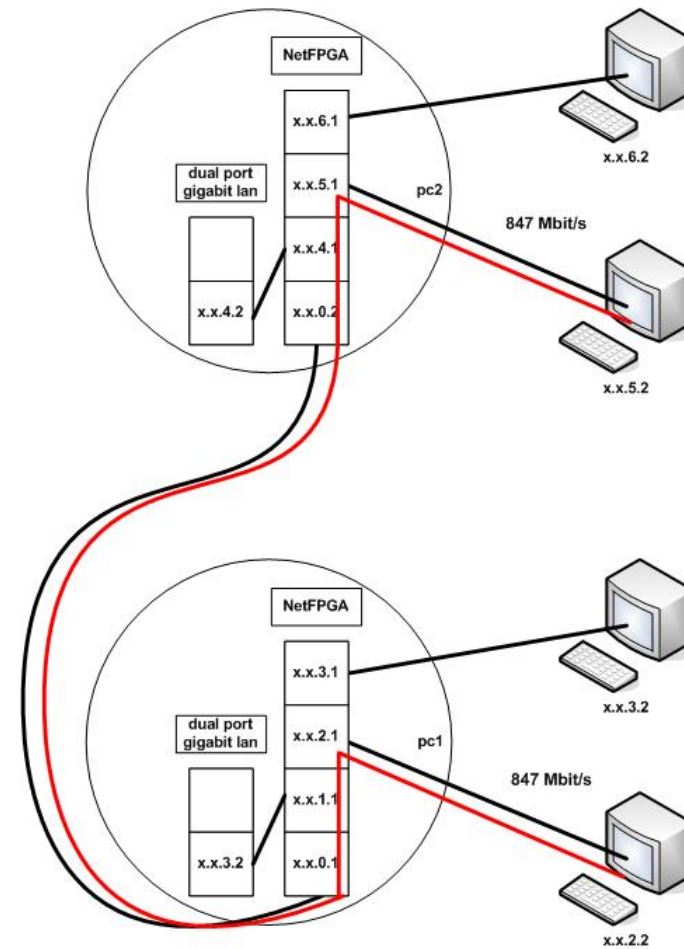


NetFPGA 대역폭 측정 실험

❖ Reno TCP

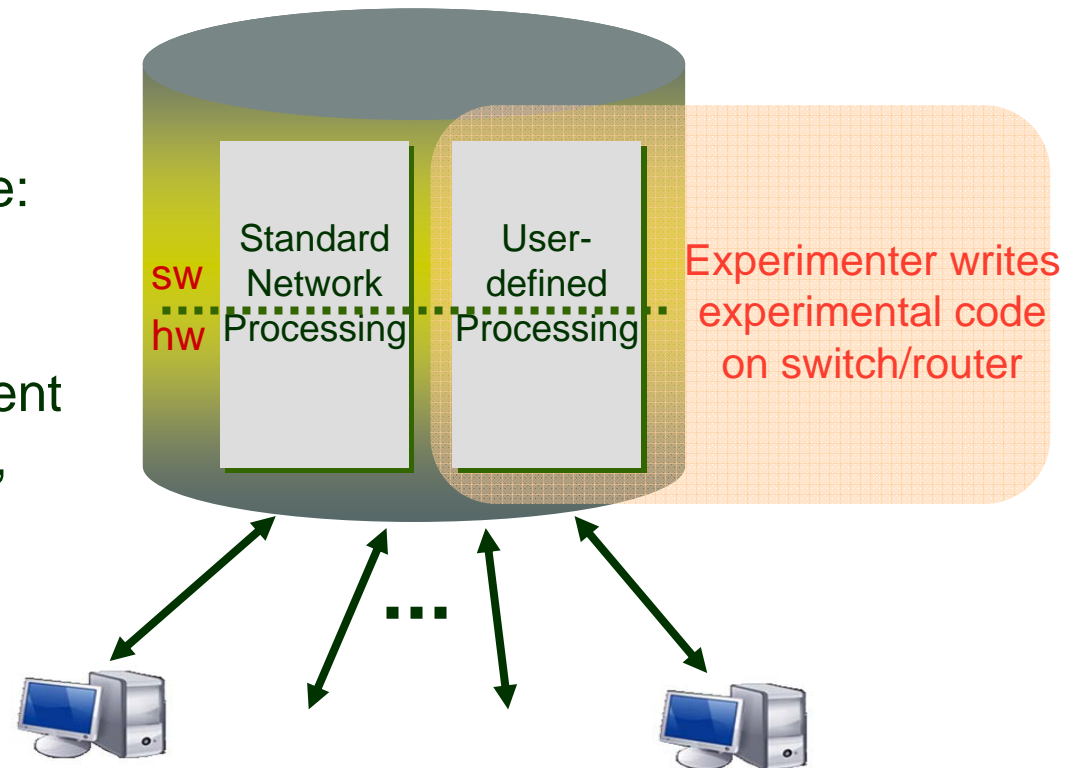


❖ HSTCP



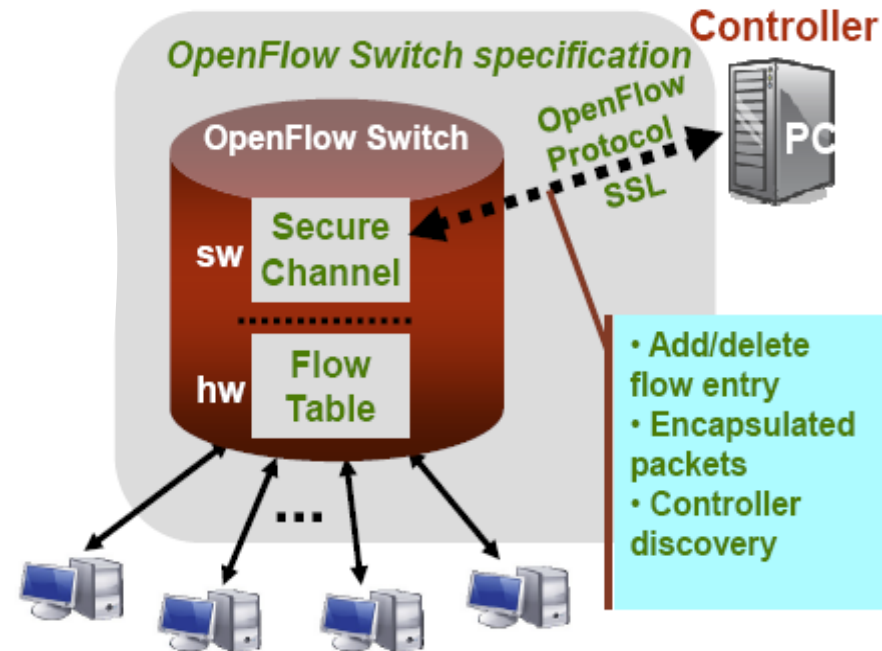
OpenFlow switching

- ❖ Isolation: Regular production traffic untouched
- ❖ Virtualized and programmable: Different flows processed in different ways
- ❖ Open development environment for all researchers (e.g. Linux, Verilog, etc).
- ❖ Flexible definitions of a flow
 - Individual application traffic
 - Aggregated flows
 - Alternatives to IP running side-by-side
 - ...



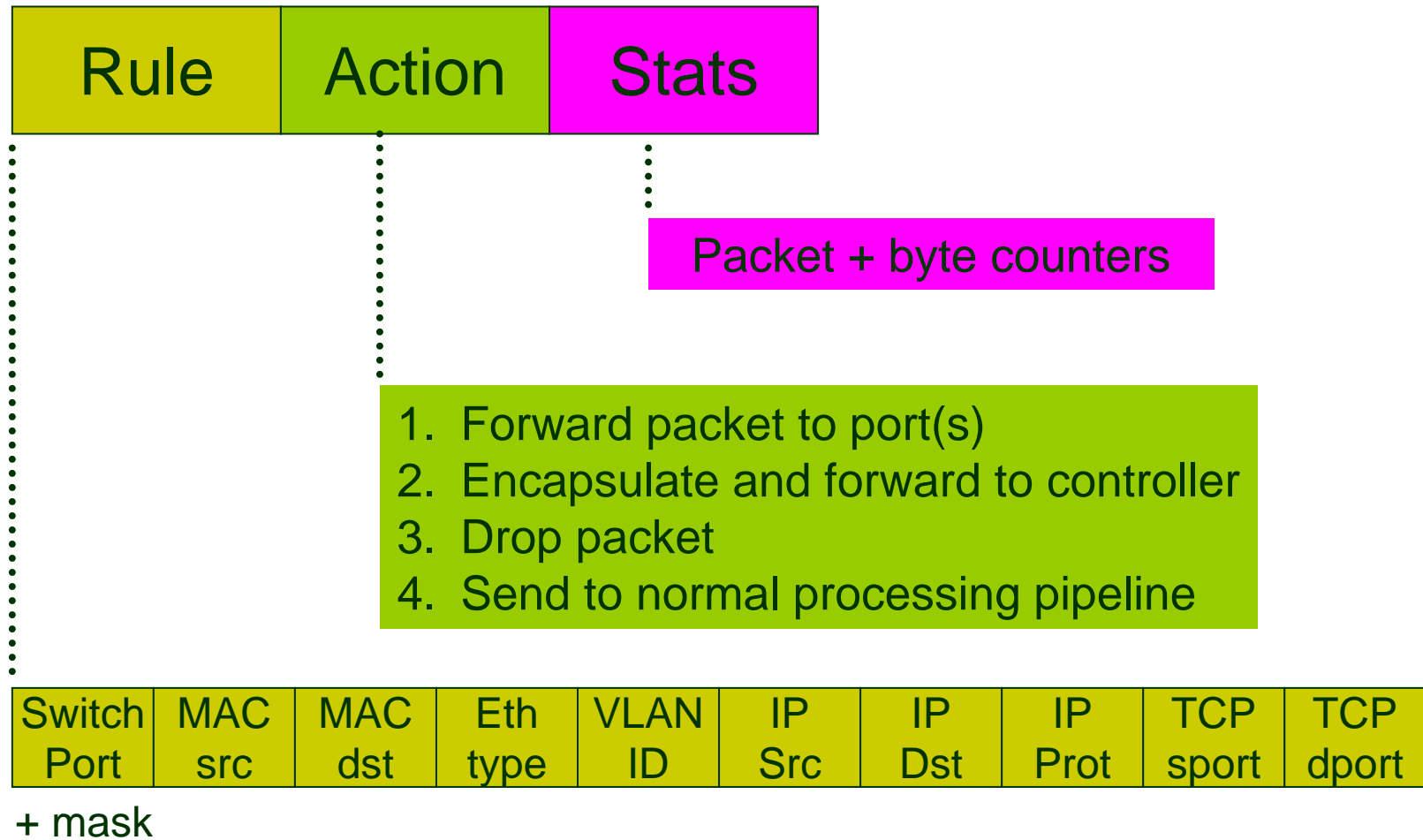
OpenFlow components

- ❖ OpenFlow Controller
- ❖ OpenFlow protocol
- ❖ Secure channel
- ❖ Flow Table
 - Ingress port
 - Ethernet src/dst addresses
 - Ethernet type
 - VLAN tag
 - IP src/dst addresses
 - IP protocol
 - TCP/UDP src/dst ports



Flow Table Entry

“Type 0” OpenFlow Switch





OpenFlow protocol

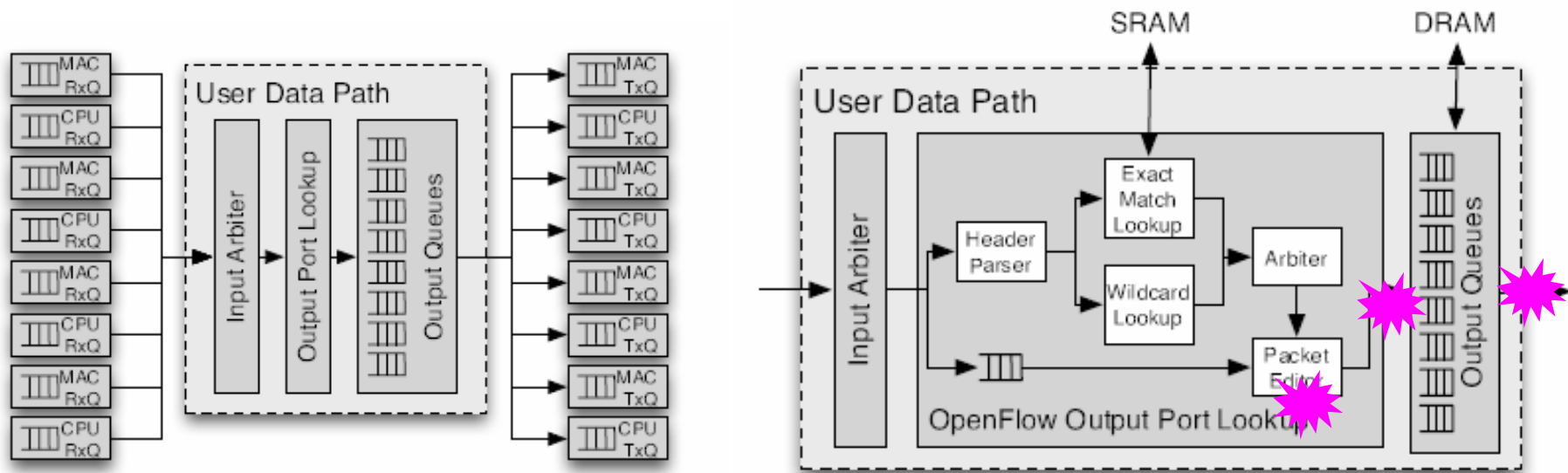
- ❖ “<http://www.openflowswitch.org/documents/openflow-spec-v0.8.9.pdf>”
- ❖ Controller-to-Switch Messages
 - Handshake messages
 - ❑ OFPT_FEATURES_REQUEST/OFTP_FEATURES_REPLY
 - Switch configuration messages
 - ❑ OFPT_SET_CONFIG/OFTP_GET_CONFIG_REQUEST/OFTP_GET_CONFIG_REPLY
 - Modify Flow messages
 - ❑ OFPT_FLOW_MOD/OFTP_PORT_MOD
 - Read state messages
 - ❑ OFPT_STATS_REQUEST/OFTP_STATS_REPLY
 - Send packet message
 - ❑ OFPT_PACKET_OUT
- ❖ Asynchronous Messages
 - OFPT_PACKET_IN/OFTP_FLOW_EXPIRED/OFTP_PORT_STATUS/OFTP_ERROR_MSG
- ❖ Symmetric messages
 - HELLO/ECHO_REQUEST/ECHO_REPLY



Using OpenFlow on NetFPGA

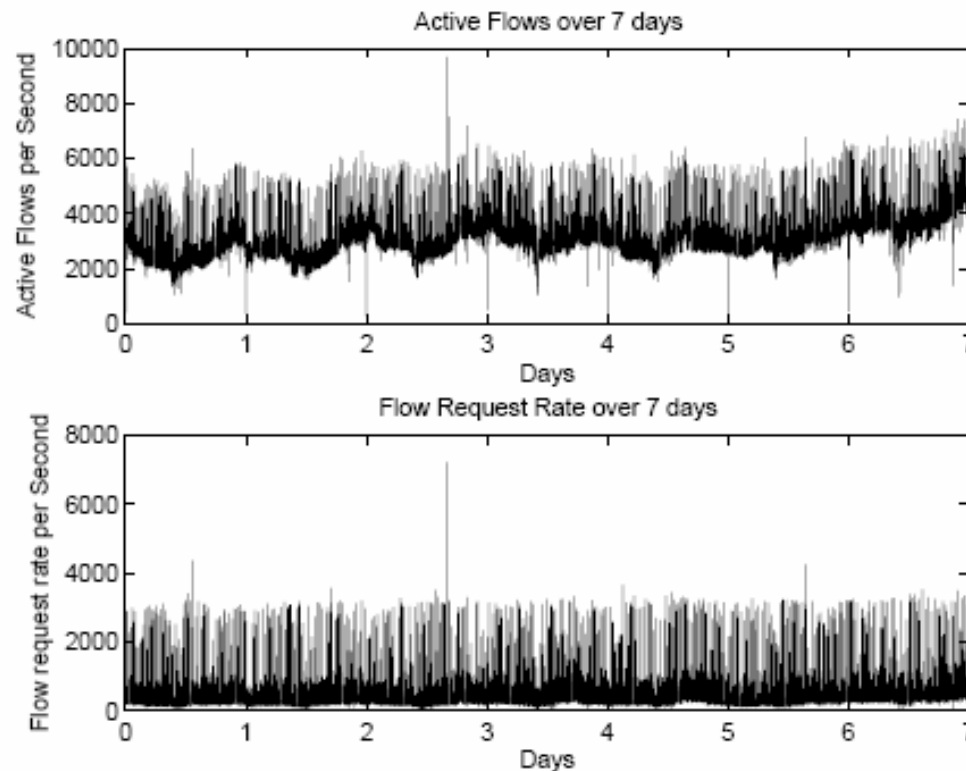
❖ Implementation

- Output port lookup module 확장



Performance Results

- ❖ The maximum number of flows active ≤ 10000 flows
 - fit very easily into the NetFPGA OpenFlow switch's 32,000-entry exact match table.
- ❖ The maximum new flow request rate seen in the Stanford EE/CS network ≤ 7000 flows/second,
 - within the capabilities of unoptimized NetFPGA OpenFlow switch implementation.



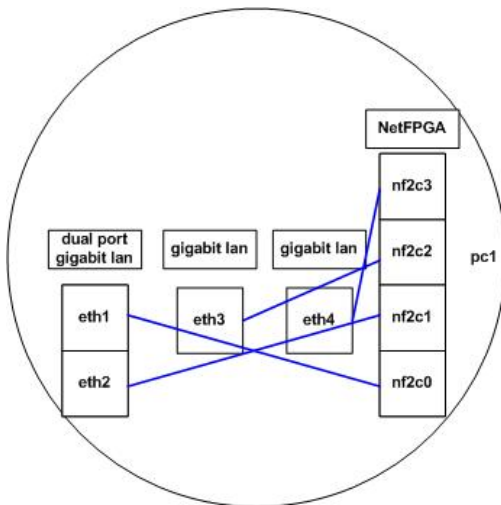
NetFPGA OpenFlow switch 설치

❖ Obtain tarball

- http://netfpga.org/beta/distributions/netfpga_openflow_switch.0_8_9-1.tar.gz

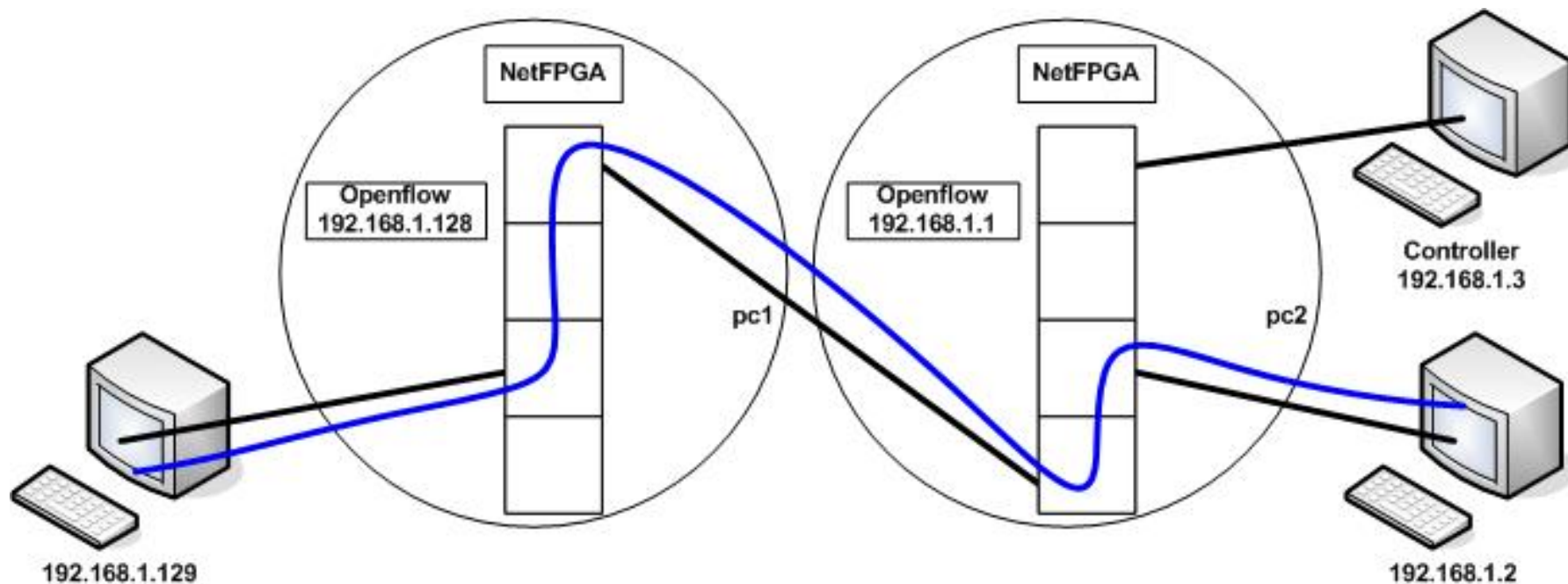
❖ Regression Tests

- Reprogram the CPCI
- Load openflow_switch bitfile
- `nf21_regression_test.pl --project openflow_switch` 실행



```
[root@localhost /]# nf21_regress_test.pl --project openflow_switch
Root directory is /root/NF2
Running tests on project 'openflow_switch'...
Running global setup... PASS
Running test 'test_exact_simple1'... PASS
Running test 'test_exact_simple2'... PASS
Running test 'test_wildcard_simple1'... PASS
Running test 'test_wildcard_simple2'... PASS
Running global teardown... PASS
```

NetFPGA OpenFlow switch 테스트



NetFPGA OpenFlow switch 테스트

❖ Controller 실행

- `./controller -v tcp:6633`

❖ Openflow switch 실행 순서

- Download OpenFlow bitfile.
 - ❑ `nf2_download openflow_switch.bit`
- Insert OpenFlow kernel module and NetFPGA kernel module
 - ❑ `/sbin/insmod openflow_mod.ko`
 - ❑ `/sbin/insmod hwtbl_nf2_mod.ko`
- Setup datapath and interface
 - ❑ `dpctl adddp nl:1`
 - ❑ `dpctl addif nl:1 nf2cX`
 - ❑ `dpctl show nl:1`
 - ❑ `ifconfig of0 192.168.1.X netmask 255.255.255.0`
 - ❑ `secchan nl:1 tcp:controller ip:port`



OpenFlow switch Datapath 설정

```
root@localhost:~  
File Edit View Terminal Tabs Help  
root@localhost:~ x root@localhost:~ x root@localhost:~ x  
[root@localhost ~]# openflow-git/utilities/dpctl adddp nl:1  
[root@localhost ~]# openflow-git/utilities/dpctl adddif nl:1 nf2c0  
[root@localhost ~]# openflow-git/utilities/dpctl adddif nl:1 nf2c1  
[root@localhost ~]# openflow-git/utilities/dpctl adddif nl:1 nf2c2  
[root@localhost ~]# openflow-git/utilities/dpctl adddif nl:1 nf2c3  
[root@localhost ~]# openflow-git/utilities/dpctl show nl:1  
features_reply (xid=0xd042b541): ver:0x97, dpid:2320d4f6d2  
n_tables:3, n_buffers:256  
features: capabilities:0x17, actions:0x3ff  
0(nf2c0): addr:00:4e:46:32:43:00, config: 0, state:0  
1(nf2c1): addr:00:4e:46:32:43:01, config: 0, state:0  
2(nf2c2): addr:00:4e:46:32:43:02, config: 0, state:0  
3(nf2c3): addr:00:4e:46:32:43:03, config: 0, state:0  
LOCAL(of0): addr:00:23:20:d4:f6:d2, config: 0, state:0  
get_config_reply (xid=0x5ea546bc): miss_send_len=128  
[root@localhost ~]# ifconfig of0 192.168.1.1 netmask 255.255.255.0  
[root@localhost ~]#
```

```
root@localhost:~  
File Edit View Terminal Tabs Help  
root@localhost:~ x root@localhost:~ x root@localhost:~ x root@localhost:~ x  
[root@localhost ~]# secchan nl:1 tcp:192.168.1.3:6633  
Feb 19 06:26:09|00001|secchan|WARN|OpenFlow reference implementation version 0.8  
.9~2  
Feb 19 06:26:09|00002|secchan|WARN|OpenFlow protocol version 0x97  
Feb 19 06:26:09|00003|rconn|WARN|nl:1: connecting...  
Feb 19 06:26:09|00004|rconn|WARN|nl:1:0: connecting...  
Feb 19 06:26:09|00005|rconn|WARN|tcp:192.168.1.3:6633: connecting...  
Feb 19 06:26:09|00006|rconn|WARN|nl:1: connected  
Feb 19 06:26:09|00007|rconn|WARN|nl:1:0: connected  
Feb 19 06:26:09|00008|port_watcher|WARN|Datapath id is 002320d4f6d2  
Feb 19 06:26:09|00009|port_watcher|WARN|Identified data path local port as "of0"  
.  
Feb 19 06:26:09|00010|rconn|WARN|tcp:192.168.1.3:6633: connected  

```



OpenFlow switch 'ping' 테스트

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# ping 192.168.1.129 -c 3  
PING 192.168.1.129 (192.168.1.129) 56(84) bytes of data.  
64 bytes from 192.168.1.129: icmp_seq=1 ttl=64 time=4.69 ms  
64 bytes from 192.168.1.129: icmp_seq=2 ttl=64 time=0.261 ms  
64 bytes from 192.168.1.129: icmp_seq=3 ttl=64 time=0.196 ms  
  
--- 192.168.1.129 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2000ms  
rtt min/avg/max/mdev = 0.196/1.718/4.698/2.107 ms  
[root@localhost ~]#  
  
root@localhost:~  
File Edit View Terminal Tabs Help  
root@loca... x root@loca... x root@loca... x root@loca... x root@loca... x root@loca... x  
[root@localhost ~]# dpctl show nl:1  
features_reply (xid=0x3ffb410): ver:0x97, dpid:2320d4f6d2  
n_tables:3, n_buffers:256  
features: capabilities:0x17, actions:0x3ff  
0(nf2c0): addr:00:4e:46:32:43:00, config: 0, state:0  
1(nf2c1): addr:00:4e:46:32:43:01, config: 0, state:0  
2(nf2c2): addr:00:4e:46:32:43:02, config: 0, state:0  
3(nf2c3): addr:00:4e:46:32:43:03, config: 0, state:0  
LOCAL(of0): addr:00:23:20:d4:f6:d2, config: 0, state:0  
get_config_reply (xid=0x71b6e177): (sending flow expirations) miss_send_len=128  
[root@localhost ~]#  
  
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# dpctl dump-flows nl:1  
stats_reply (xid=0x3bde0ccd): flags=none type=1(flow)  
duration=2853s, table_id=0, priority=65535, n_packets=1, n_bytes=15390, idle_timeo  
ut=0,hard_timeout=0,tcp,in_port=3,dl_vlan=0xffff,dl_src=00:30:48:43:b4:05,dl_d  
st=00:23:20:e5:0d:84,nw_src=192.168.1.3,nw_dst=192.168.1.128,tp_src=6633,tp_dst=3  
6658,actions=output:0  
duration=2853s, table_id=0, priority=65535, n_packets=1, n_bytes=28756, idle_ti  
meout=0,hard_timeout=0,tcp,in_port=0,dl_vlan=0xffff,dl_src=00:23:20:e5:0d:84,dl_d  
st=00:30:48:43:b4:05,nw_src=192.168.1.128,nw_dst=192.168.1.3,tp_src=36658,tp_dst=  
6633,actions=output:3  
duration=3002s, table_id=0, priority=65535, n_packets=417, n_bytes=30226, idle_  
timeout=0,hard_timeout=0,tcp,in_port=65534,dl_vlan=0xffff,dl_src=00:23:20:c5:94:4  
0,dl_dst=00:30:48:43:b4:05,nw_src=192.168.1.1,nw_dst=192.168.1.3,tp_src=33291,tp_  
dst=6633,actions=output:3  
duration=3002s, table_id=1, priority=65535, n_packets=213, n_bytes=16014, idle_  
timeout=0,hard_timeout=0,tcp,in_port=3,dl_vlan=0xffff,dl_src=00:30:48:43:b4:05,dl_  
dst=00:23:20:c5:94:40,nw_src=192.168.1.3,nw_dst=192.168.1.1,tp_src=6633,tp_dst=3  
3291,actions=LOCAL  
duration=3002s, table_id=1, priority=65535, n_packets=1, n_bytes=60, idle_timeo  
ut=0,hard_timeout=0,arp,in_port=3,dl_vlan=0xffff,dl_src=00:30:48:43:b4:05,dl_d  
st=00:23:20:c5:94:40,nw_src=0.0.0.0,nw_dst=0.0.0.0,nw_proto=0,tp_src=0,tp_dst=0,acti  
ons=LOCAL  
[root@localhost ~]#
```

OpenFlow switch 'ping' 테스트

The screenshot shows a Wireshark capture of network traffic. The main pane displays a list of captured packets. Packet 13 is selected, showing an OpenFlow Set Config message. The details pane for this packet is expanded to show the Switch Configuration section, specifically the Flags field. The flags indicate that flow expirations are sent to the controller (Yes) and that there is no special handling for IP fragments. The maximum bytes for a new flow to be sent to the controller is 128.

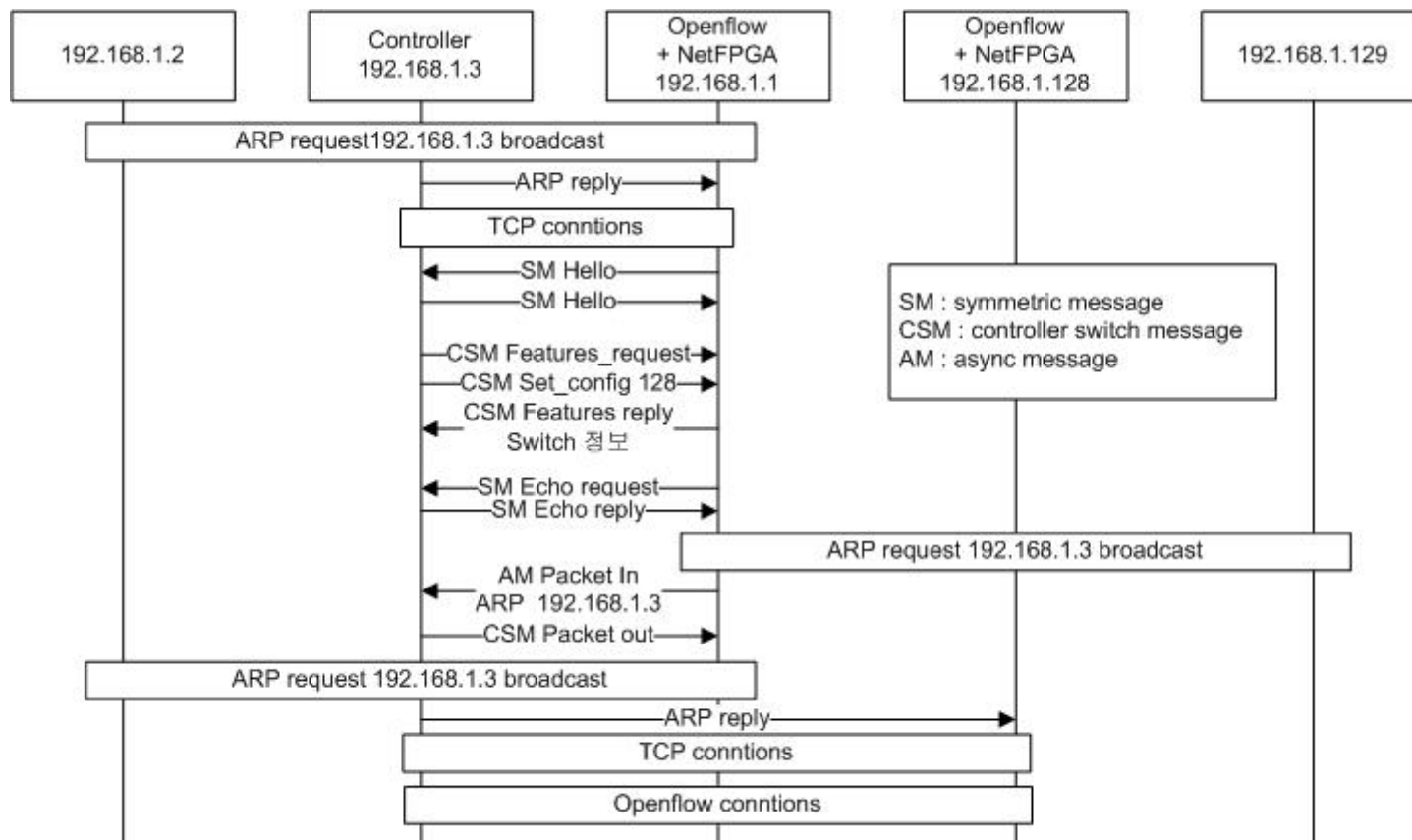
No.	Time	Source	Destination	Protocol	Info
10	0.001245	192.168.1.3	192.168.1.1	OFPP	Features Request (CSM) (88)
11	0.001250	192.168.1.1	192.168.1.3	TCP	45570 > 6633 [ACK] Seq=9 Ack=17 Win=5888 Len=0 TSV=10
12	0.001316	192.168.1.1	192.168.1.3	OFPP	Features Reply (CSM) (272B)
13	0.001361	192.168.1.3	192.168.1.1	OFPP	Set Config (CSM) (12B)
14	0.040967	192.168.1.1	192.168.1.3	TCP	45570 > 6633 [ACK] Seq=281 Ack=29 Win=5888 Len=0 TSV=10
15	0.041271	192.168.1.3	192.168.1.1	TCP	6633 > 45570 [ACK] Seq=29 Ack=281 Win=6880 Len=0 TSV=10
16	15.001085	192.168.1.1	192.168.1.3	OFPP	Echo Request (SM) (8B)

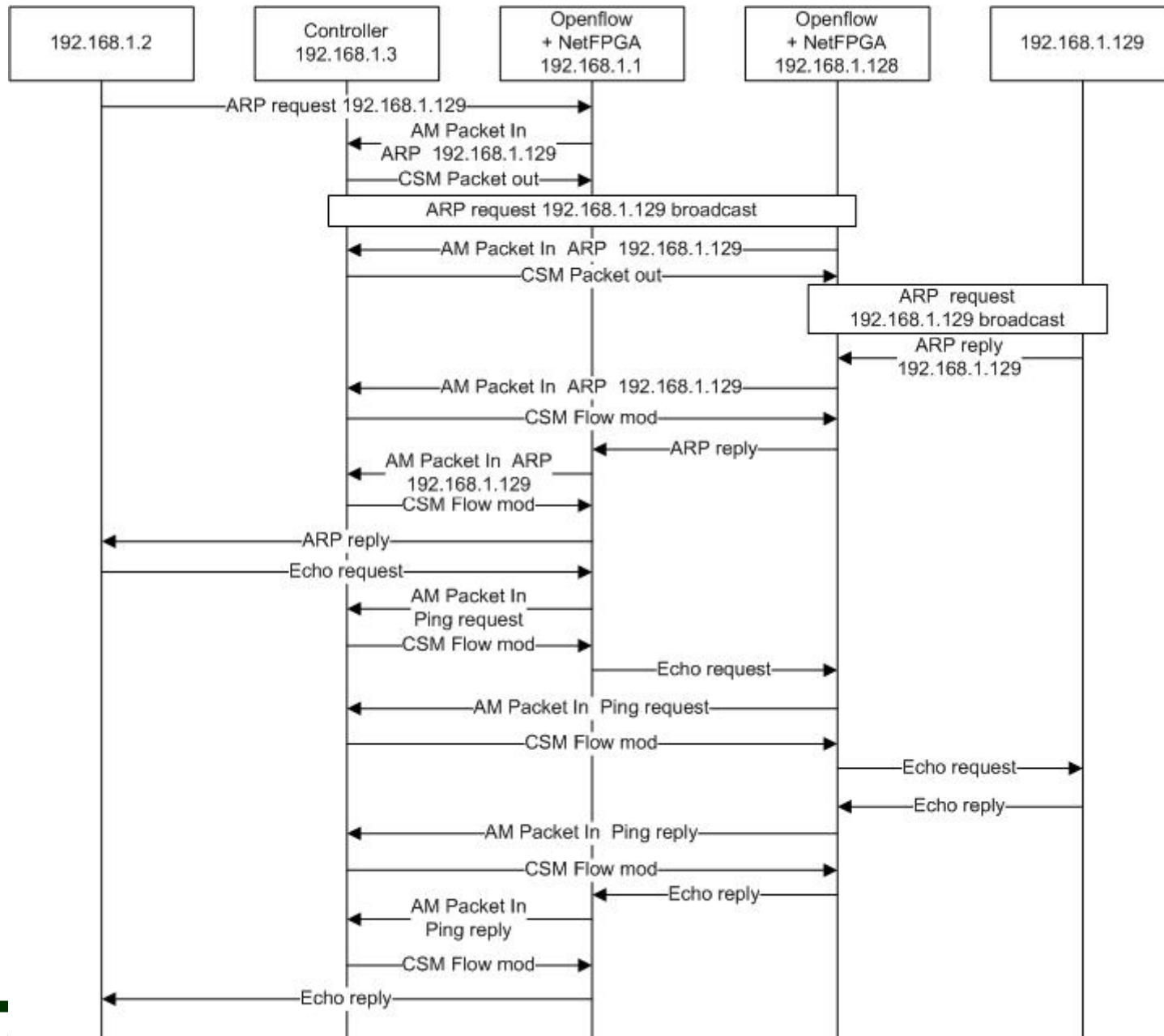
Frame 13 (78 bytes on wire, 78 bytes captured)

- Ethernet II, Src: Supermic_43:b4:05 (00:30:48:43:b4:05), Dst: NiciraNe_d4:f6:d2 (00:23:20:d4:f6:d2)
- Internet Protocol, Src: 192.168.1.3 (192.168.1.3), Dst: 192.168.1.1 (192.168.1.1)
- Transmission Control Protocol, Src Port: 6633 (6633), Dst Port: 45570 (45570), Seq: 17, Ack: 9, Len: 12
- OpenFlow Protocol
 - Header
 - Switch Configuration
 - Flags
 - ...1 = Send flow expirations: Yes (1)
 - ...00 = Handling of IP fragments: No special fragment handling (0)
 - Max Bytes of New Flow to Send to Controller: 128



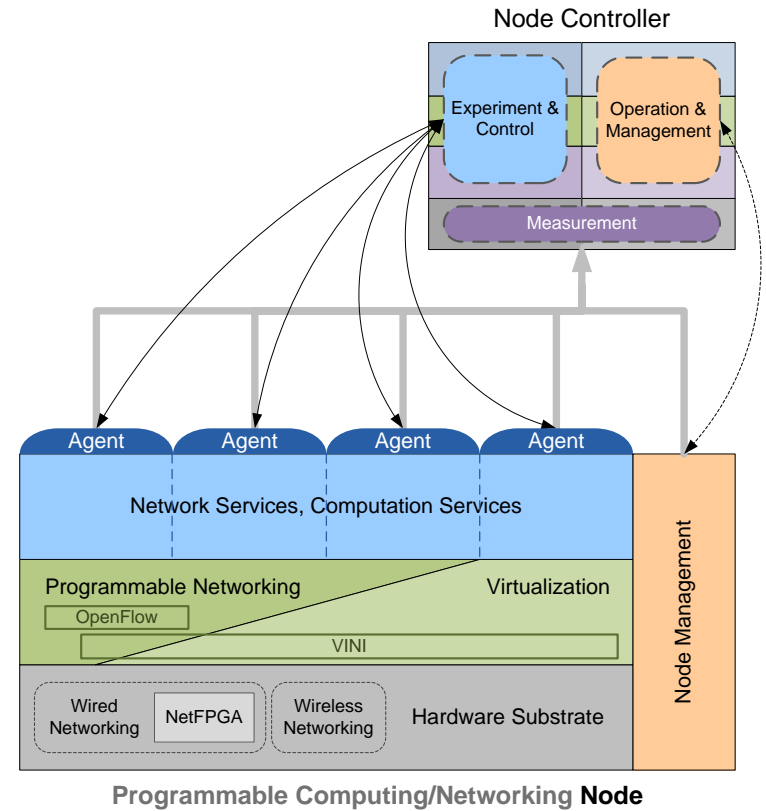
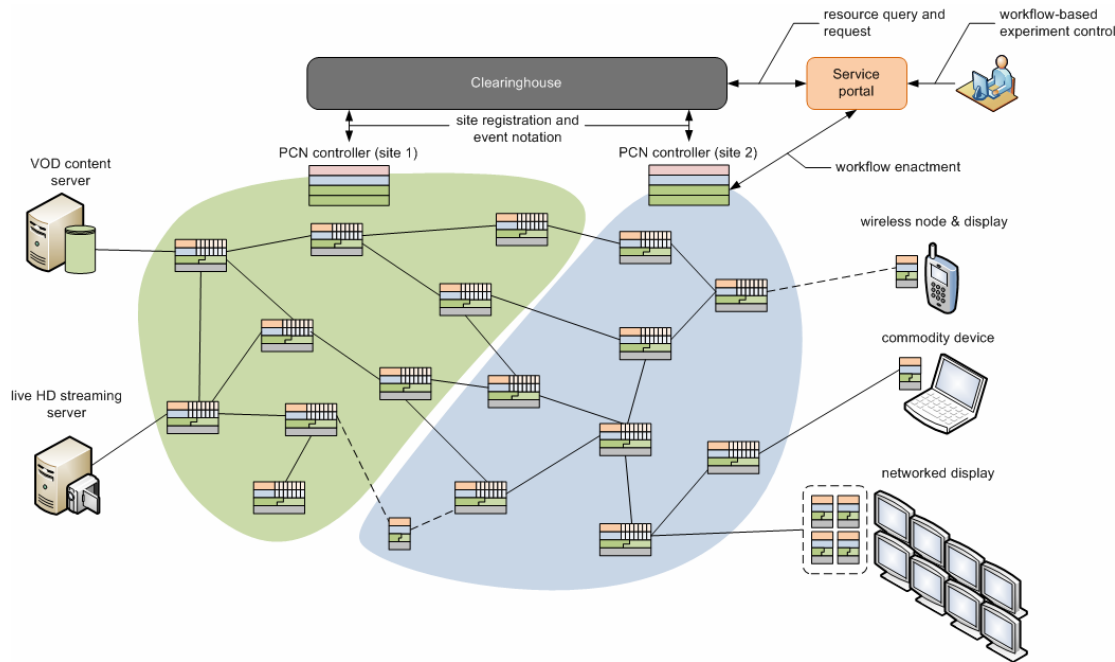
OpenFlow switch 'ping' 테스트





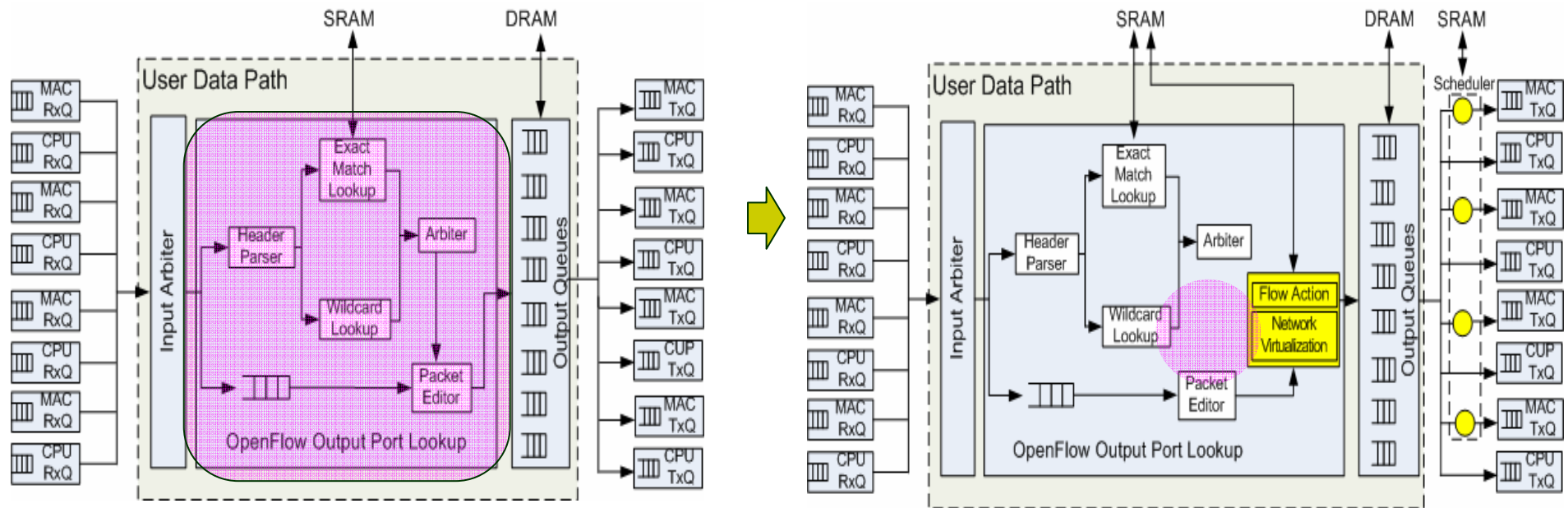
향후 연구

- ❖ “미래인터넷 인프라를 위한 PC기반 네트워크 가상화 노드 및 테스트베드 구축” 연구



향후 연구

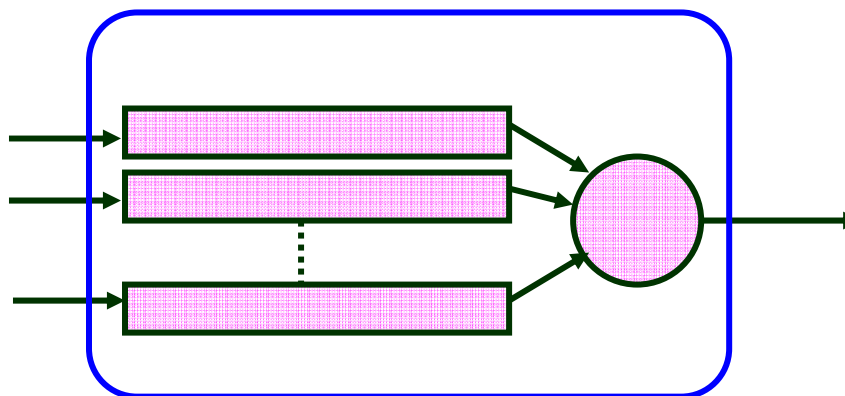
- ❖ Programmable 네트워크 장치의 다중 네트워크 지원 자원관리 및 가상화 기술 개발
 - 자원관리 scheduler (DRR, WFQ 등)
 - Packet editor 확장 (NAT, Other protocols)
 - Openflow protocol 확장



향후 연구

❖ Scheduler 구현

- Output queue 확장
- DRR, WFQ, SCFQ 등



Output Queue

❖ OpenFlow protocol 확장

- Flow entry 및 message format 수정





References

- ❖ NetFPGA Guide document,
 - <http://netfpga.org/netfpgawiki/index.php/Guide>
- ❖ Nick McKeown, et al., “OpenFlow: Enabling Innovation in Campus Networks”, Whitepaper,
 - <http://www.openflowswitch.org//documents/openflow-wp-latest.pdf>
- ❖ Brandon Heller, “OpenFlow Switch Specification”,
 - <http://www.openflowswitch.org/documents/openflow-spec-v0.8.9.pdf>
- ❖ Jad Naous, et al., “Implementing an OpenFlow Switch on the NetFPGA platform”, ACM ANCS’08, Nov. 2008