

## IXDP2850 기반 트래픽 모니터링

송일섭<sup>o</sup> 정해진 권택근  
충남대학교 컴퓨터공학과

{issong<sup>o</sup>, hjjung, tgkwon}@ce.cnu.ac.kr

### Traffic Monitoring Based on IXDP2850

Ilsoop Song<sup>o</sup> Haejin Jung Taeckgeun Kwon  
Dept. of Computer Engineering, Chungnam National University

#### 요 약

현재 라우터에서의 트래픽 모니터링은 라우터의 성능저하를 최소화하기 위하여 샘플링을 통한 트래픽 모니터링을 사용한다. 기존의 샘플링을 통한 트래픽 모니터링 대신 고속 라우터의 성능을 그대로 유지하면서 모든 트래픽을 모니터링을 하고, 또한 다양하고 급격히 변화하는 트래픽에 대해서 보다 능동적으로 모니터링 할 수 있는 방법을 네트워크 프로세서를 통하여 가능성을 제시한다.

#### 1. 서 론

최근에 라우터들은 트래픽 모니터링을 제공하고 있으며 모니터링 된 트래픽 플로우의 통계 자료들을 다루기 위하여 SNMP (Simple Network Monitoring), RMON (Remote Monitoring), 그리고 NetFlow를 마련하고 있다. 그러나 전통적인 트래픽 모니터링의 기능들은 미리 정해져 있어서 그 기능들을 수정할 수 없었다. 때문에 앞으로의 지능형 네트워크 장비에서는 이러한 전통적인 트래픽 모니터링이 적합하지 않다. 더욱이 현재의 대부분 라우터들은 트래픽 모니터링을 수행함으로써 라우터의 성능을 전체적으로 저하시키고, 실시간으로 흐르는 모든 트래픽에 대한 측정을 불가능하다. 한 예로 많은 ISPs (Internet Service Providers)의 네트워크 운영자들이 라우터의 성능을 저하를 최소화하기 위하여 샘플링 방법을 선택하고 있다.

본 논문에서는 10Gb의 고속 라우터 성능을 그대로 유지하면서 트래픽에 대한 모든 플로우들의 상태를 유지관리하며 네트워크 프로세서의 마이크로 프로그래밍을 통한 보다 유연한 트래픽 모니터링 모듈을 개발함에 있다.

본 논문의 구성은 다음과 같다.

제2장에서는 인텔 네트워크 프로세서인 IXDP2850에 대한 구조와 마이크로 엔진에 대해 설명을 하고, 제3장에서는 IXDP2850을 이용한 트래픽 모니터링 방법을 설명한다. 그리고 제4장에서는 트래픽 모니터링의 실험환경과 실험결과를 기술하고 결론 및 향후 연구과제는 제5장에서 기술한다.

#### 2. 인텔 IXDP2850 네트워크 프로세서

인텔에서는 네트워크 프로세서로 IXP1200, IXP2400,

IXP2800, IXP2850을 제공하고 있으며, 그 중 IXP2850은 IXP2800에 크립트 유닛 (Crypto Unit)을 추가한 네트워크 프로세서로 본 논문에서의 트래픽 모니터링을 위한 마이크로 코드 구현에 사용한 네트워크 프로세서다.

IXP2850 네트워크 프로세서의 내부 구조는 다음과 같다

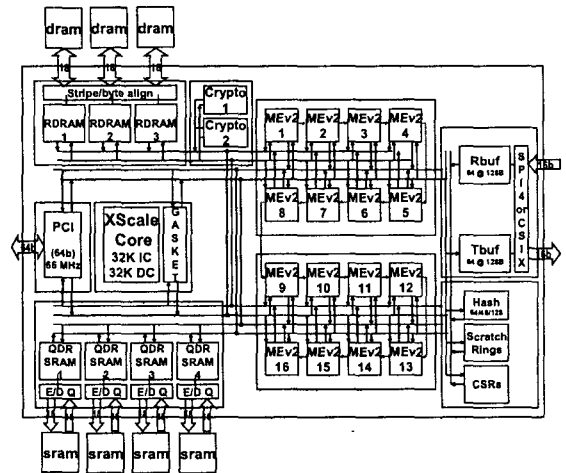


그림 1 IXDP2850 내부 구조

IXP2850은 XScale Core와 8개의 마이크로 엔진을 갖는 클러스터가 두 개 있으며 각 마이크로 엔진당 8개의 쓰레드를 갖는다. 또한 메모리의 동시 접근으로 인한 병목현상을 줄이기 위하여 각각 다른 채널을 갖는 3개의 DRAM과 4개의 SRAM이 존재한다. 그리고 MSF (Media Switch Fabric)으로부터 패킷을 받고 보내는 Rbuf (Receive Buffer)와 Tbuf (Transmit Buffer)가 있고, 이는 SPI4

또는 CSIX를 지원한다. 그 외에도 Hash, Scratch Ring, CSRs (Control and Status Resgisters) 그리고 크립토 유닛이 있다[1-3].

아래의 그림은 MEv2 (Micro Engine Version 2) 내부 구조를 보여준다.

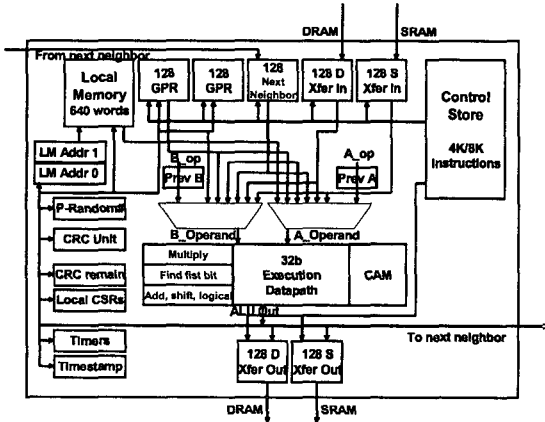


그림 2 IXP2850의 MEv2 내부 구조

MEv2에는 2개의 GPR (General Purpose Register), Next Neighbor Register, DRAM Transfer read/write Register, SRAM Transfer read/write Register 그리고 Local Memory가 존재한다. 그 외에 다수의 특별한 목적으로 쓰여지는 Register들이 있다. [그림 2]의 레지스터와 Local Memory의 숫자들은 사이즈를 의미하며, 단위는 LW (Long Word, 4bytes)이다.

본 논문의 실험에 사용된 IXP2850은 상향 진입지점 (ingress) 그리고 하향 진입지점 (egress)에 네트워크 프로세서 IXP2850 2개가 사용된다.

### 3. IXP2850 기반의 트래픽 모니터링

트래픽 모니터링을 위한 시스템은 크게 포워딩 플레인과 컨트롤 플레인으로 나눌 수 있으며, 그 전체적인 모습은 [그림 3]과 같다.

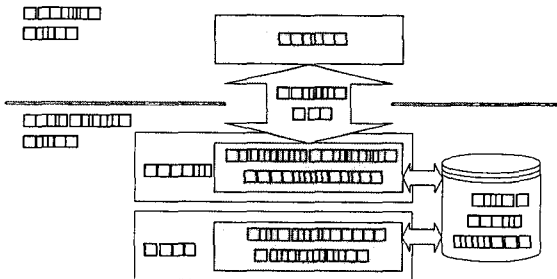


그림 3 트래픽 모니터링 시스템 구성도

본 장에서는 포워딩 플레인을 중점으로 설명한다.

#### 3.1 포워딩 플레인

포워딩 플레인은 마이크로 코드가 실행되는 MEv2 부분과 리눅스 커널이 올라간 XScale로 구분할 수 있다.

MEv2에서는 라우팅 테이블을 이용하여 패킷을 포워딩하고 플로우를 생성, 갱신하는 마이크로 코드가 실행되며, XScale에서는 마이크로 코드가 의해 관리되는 플로우들을 읽어 nProbe로 보내는 일을 한다.

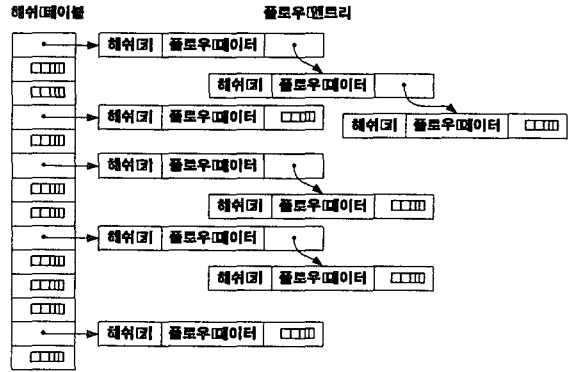


그림 4 플로우 테이블 구성도

트래픽 모니터링 마이크로 코드의 내용을 간략히 설명하면 아래와 같다[4].

- [그림 4]에서와 같이 패킷의 5-튜플의 해싱 결과를 이용하여 해쉬 테이블의 위치를 찾고, 들어온 패킷의 5-튜플과 같은 플로우가 있는지를 찾는다.
- 같은 플로우가 존재하면 플로우 엔트리의 내용을 업데이트를 한다.
- 같은 플로우를 찾지 못하면 새로운 플로우 엔트리를 생성하고 플로우 테이블에 저장한다.
- 해쉬 테이블의 엔트리에 연결된 플로우 엔트리들의 개수가 4개를 초과하면, 4개의 플로우 엔트리 중 가장 오래된 플로우 엔트리를 OFT (Old Flow Table)로 옮기고 새로운 플로우를 저장한다.

플로우 엔트리의 필드들의 내용은 아래 [표 1]과 같다.

표 1 플로우 엔트리 속성들

필드	설명
Protocol	Protocol ID (e.g. TCP, UDP, ICMP, ...) for IP packet payload
Source/destination IP address	Source and destination IP address
Source/destination port number	L4 source and destination port number (e.g. TCP ports)
Input/output port number	Ingress egress port number
TOS	Type of services field for IP header
Flags	TCP flags (OR' ed flags)
Byte counter	Number of bytes received
Packet counter	Number of packets received
First time	Timestamp for the first packet of the flow arrived
Last time	Timestamp for the last packet arrived
Current time	Current timestamp

### 3.2 컨트롤 플레인

nProbe는 pcaplib를 이용하여 IPFIX를 생성하고 분석하는 네트워크 트래픽 수집기이다. 기존의 nProbe는 nProbe가 설치된 호스트의 네트워크 카드를 통해 들어오는 패킷들을 대상으로 트래픽을 수집하나 이를 수정하여 XScale로부터 플로우 정보들을 받도록 하였다[5,6].

nProbe와 XScale 사이의 IPC는 보다 융통성 있는 IPC를 위하여 netlink 메시지를 사용하였다[7].

### 4. 트래픽 모니터링 실험 결과

본 논문에서는 [그림 5]와 같이 트래픽 모니터링 성능 실험을 위해 네트워크 성능 분석 시스템인 SmartBits 6000B를 사용하였다. IXDP2850이 10Gb 성능을 지원하는 반면, 실험에 사용한 SmartBits 6000B가 2Gb 성능 측정만을 할 수 있어서 마이크로 코드의 트래픽 모니터링 블록을 5번 반복 실행하여 10Gb의 성능을 측정할 수 있게 하였다.

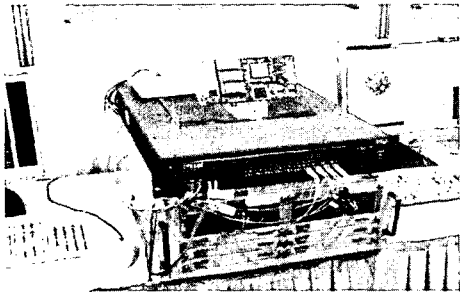


그림 5 실험 환경

패킷 크기를 64바이트, 128바이트, 256바이트 그리고 512바이트 별로 패킷 처리율을 측정하고, 또한 각 패킷 크기당 70%, 80%, 90% 그리고 100% 부하 (load)로 패킷 지연시간을 측정하였다.

패킷 처리율에서는 [그림 6]에서와 같이 패킷 크기가 64bytes가 91.30%의 처리율을 보이고 그 외의 모든 패킷 크기에서 100%의 처리율을 보였다.

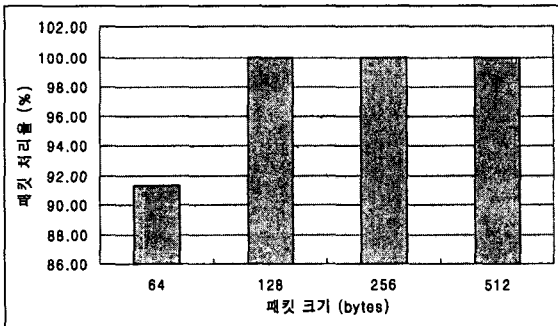


그림 6 패킷 처리율

그리고 패킷 지연시간의 결과에서는 패킷 크기가 64바이트에 100%의 로드를 주었을 때 53us의 지연시간을 보이고, 그 외에는 10us 내외의 지연시간을 보였다. 패킷 지연시간의 자세한 결과는 [그림 7]과 같다.

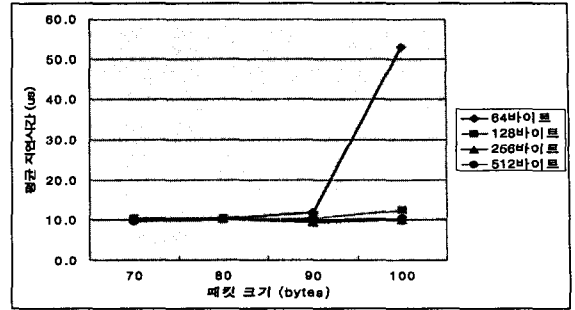


그림 7 패킷 지연시간

### 5. 결론 및 향후 연구과제

인텔 IXD2850 네트워크 프로세서를 이용하여 10Gb의 고속 라우터의 성능을 거의 모든 트래픽을 모니터링 할 수 있음을 보였다. 또한 트래픽 모니터링의 내용을 마이크로 코드를 통해 보다 능동적으로 대처할 수 있다

실제 네트워크 트래픽에서는 가장 작은 패킷이 40바이트임을 감안할 때 10Gb의 라우터 성능을 보장할 수 없다. 그 대안으로 트래픽 모니터링 마이크로 블록을 하나의 마이크로 엔진에서 두 개의 마이크로 엔진에 할당하여 16개의 쓰레드를 파이프 라인으로 구현하면 10Gb의 라우터 성능을 보장할 수 있을 것이다.

### 참고문헌

- [1] Intel, Intel® IXP2850 Network Processor, [http://www.intel.com/design/network/products/np\\_family/ixp2850.htm](http://www.intel.com/design/network/products/np_family/ixp2850.htm), 2004.
- [2] B. Carlson, Intel® Internet Exchange Architecture and Applications: A Practical Guide to IXP2XXX Network Processors, Intel Press, 2003.
- [3] Intel, Intel® IXP2800/IXP2850 Network Processor: Hardware Reference Manual, 2002.
- [4] E. J. Johnson and A. R. Kunze, IXP2400/2800 Programming: The Complete Microengine Coding Guide, Intel Press, 2003.
- [5] IETF, IP Flow Information Export (IPFIX), <http://www.ietf.org/html.charters/ipfix-charter.html>, 2004
- [6] L. Deri, Open Source NetFlow Probe for Gigabit Networks, <http://www.terena.nl/conferences/tnc2003/programme/slides/s8b2.pdf>, 2004.
- [7] J. Salim, H. Khosravi, A. Kleen, A. Kuznetsov, Linux Netlink as an IP Services Protocol, RFC 3549, 2003