고속 패킷 처리를 위한 네트워크 프로세서 아키텍처 비교

진현정, 손경덕, 김화종
강원대학교 전자공학과

Comparison of Network Processor Architecture for High Speed Packet Processing

Hyun Joung Jin, Kyung Duck Son, Hwa Jong Kim
Dept. Electronics Engineering, Kangwon National University

Abstract - 네트워크 프로세서는 주로 라우터에 위치하여 데이터의 트래픽을 관리를 하였다. 네트워크 프로세서는 데이터의 빠른 전송을 위해 점차 발달하였으며, 종류 또한 다양해졌다. 본 논문에서는 네트워크 프로세서의 아키텍처가 빠른 패킷 처리를 위해 어떻게 발전하였는지 알아본다. 또한 기존의 네트워크 프로세서들이 어디서 동작하고, 어떤 아키텍처를 썼는지를 기준으로 분석하고, 분석 결과를 분류하였다. 네트워크 프로세서는 router나 line card 등의 다양한 위치에 위치하며, OSI 계층의 사용범위가 다양함을 보여주었다.

1. 서 론

인터넷이 발달함에 따라 새로운 프로토콜과 네트워크 데이터 타입의 처리가 요구되었다. 또한 인터넷이 더욱 복잡해지면서 속도가 빨라짐에 따라 새로운 장비와 업그레이드가 필요하게 되었다. 네트워크 프로세서는 wire speed의 프로세스를 요구하는 높은 성능의 하드웨어로 패킷들을 전달하는 동안 시스템 유연성을 제공한다. 네트워크 프로세서는 System-on-Chip(SoC) 기술을 기반으로 하는데, 이 기술은 general-purpose processor보다 데이터의 통신을 더 효율적으로 처리할 수 있게 한다[1].

네트워크 프로세서는 디지털 시그널로 데이터를 컨버팅하는 물리적 계층 기능, 디바이스 안에서 트래픽을 지정하는 switching fabric 컨트롤 기능, 각종 프로토콜의 프로세싱을 다루는 패킷 프로세싱 기능, 대부분 네트워크 프로세싱 유닛들을 위해 시스템 컨트롤이나 호스트 프로세싱 기능을 가지고 있다[1].

네트워크 프로세서의 디자인은 general RISC-based architecture, augment RISC architecture, network-specific processor등 세 가지 타입으로 나누어진다[2].

본 논문에서는 기존의 네트워크·프로세서들의 구조를 조사하였으며 이들이 어떤 곳에서 사용되는지, 어느 계층에서 작동하는지, 어떤 아키텍처를 적용하는지를 분석하였다. 최신의 네트워크 프로세서보다는 널리 사용되는 프로세서 위주로 분류하였다[3]-[16].

2. 네트워크 프로세서 아키텍처

2.1 일반적인 RISC기반의 아키텍처

일부 회사들이 네트워크 프로세서를 디자인할 때 하나의 칩에 규격품인 RISC(Reduced Instruction Set Circuit) 프로세서들을 통합하고자 시도하였다. RISC는 CISC(Complex Instruction Set Circuit)보다 간단한 instruction을 사용하여 칩의 복잡성을 줄이고자 시도한 것이다. RISC의 특징으로는 작은 instruction set, load/store 아키텍처, 고정된 길이의 코딩과 하드웨어 디코딩이 있다. 또한 RISC는 싸이클 당 하나의 instruction을 통해 processor의 throughput이 발생하는 특징이 있다.

RISC 장비는 마이크로코드 컨벤션 레이어를 통하지 않기 때문에 instruction을 더 빨리 실행하지만, RISC 기반의 네트워크 프로세서는 복잡한 일을 수행할 때 시간이 걸리는 단점이 있다. RISC는 높은 스피드의 제품을 위해 병렬적으로 배치를 하나 아키텍처는 여전히 RISC의 throughput을 가지고 있다. 더구나 칩의 크기나 시스템의 복잡성을 고려치 않고 합칠 수 있는 RISC의 수는 제한되어 있다.

2.2 가속된 RISC 기반의 아키텍처

가속된 RISC 기반의 네트워크 프로세서는 RISC에 네트워크 기능을 맞추고 하드웨어 가속기를 추가한 것으로 프로세싱 속도를 빠르게 하기 위해 제안되었다. 하드웨어 가속기(accelerator)는 프레임 복사를 wire speed로 할 수 있기 때문에 성능을 높이는 역할을 한다. 그러나 가속기는 자체적으로 유연성이 부족하고 프로그래밍 할 수 없는 단점이 있다. 결국 빨라진 것을 제외하면 RISC의 단점을 유지하고 있다[2].

이 문제를 해결하기 위해 RISC와 ASIC를 합친 아키텍처를 만들었으며, 이 아키텍처는 RISC가 핵심 프로세서로 작동하고 그 외의 일들은 ASIC에 할당하는 형태의 아키텍처이다. ASIC 아키텍처를 적용하면 임베디드 형식의 칩이 되고, hard-wired의 빠른 스피드를 제공하지만, ASIC 임베디드 형식이어서 새로운 기능을 지원하기 위해 업데이트하지 못하며 새로운 요구에 적절하게 대응할 수 없는 유연성의 부족이라는 단점이 있다. 또한 고비용과 장기간의 설계 기간의 단점도 있다[5].

2.3 네트워크 특수 프로세서

네트워크 특수 프로세서는 현재의 네트워크 제품들이 필요로 하는 프로토콜 프로세싱을 지원한다. 즉, 예전에는 네트워크 제품들은 데이터를 주고 받는 것에만 중점을 두면 되었으나 현재의 네트워크 제품들은 급격히 증가한 데이터들의 빠른 처리들은 물론, 보안 및 새롭게 등장한 프로토콜들을 지원할 수 있어야 한다.

네트워크 특수 프로세서는 각각의 네트워크에 관련된 일을 하는 고속의 소형 코어 프로세서들을 통합한 구조를 갖는다.

네트워크 특수 프로세서는 상당한 수의 port를 Gigabit와 Terabit 속도로 처리함으로써 패킷 프로세싱 성능을 높일 수 있다. 특히, network specific processor를 사용하여 instruction set과 data path를 최적화함으로써 고속의 패킷 프로세싱이 가능하게 되었다. task 지향적인 프로세서는 특정 네트워킹 기능을 가지고 있어, 간결한 instruction set을 가지고 일을 할 수 있다. 예를 들어, 동일한 패킷을 처리한다면 RISC보다 1/10의 명령 수를 가지고 일을 처리할 수 있다[2].

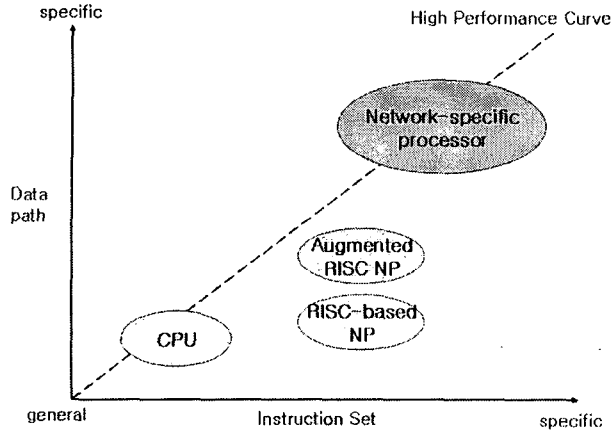

그림 1. 네트워크 프로세서 성능 그래프

3. 네트워크 프로세서 디자인

3.1 네트워크 프로세서 디자인 개요

네트워크 프로세서는 매우 복잡한 프로세싱 디바이스 중 하나이다.

네트워크 프로세서를 다자인할 때 패킷 프로세싱에 중점을 두고 디자인하여야 한다. 또 네트워크 프로세서는 어떤 특정 기능을 위해 시스템 아키텍처에 의존하게 되는데 특히 시스템과 네트워크 프로세서 사이에서의 프로세싱 방법에 중점을 둔다.

네트워크 프로세서는 특정 프로토콜이나 프로토콜 일부를 처리하기 위해 디자인하는 것이 아니라, 임의의 프로토콜 프로세싱을 높은 속도로 처리하기 위해 최소한의 set으로 디자인하여야 한다[13].

네트워크 패킷 처리의 복잡함을 줄이기 위해 하드웨어 디자이너들은 종종 divide-and-conquer 방법으로 접근하고 있으며, 그 중 한 방법으로 Ingress and Egress processing 방법을 사용한다. 이 방법은 패킷 프로세싱을 할 때 작업을 관련된 두 그룹으로 나누어 처리하는 방법이며, 기본적으로 패킷들이 switching fabric을 통과할 때 ingress와 egress가 발생한다. ingress 프로세싱은 주로 패킷들이 들어올 때 이들을 처리하기 쉽게 하기 위해 에러확인, 트래픽 관리 등을 하고 egress 프로세싱은 패킷들이 나갈 때 하는 일인 segmentation, 헤더 수정, 에러 확인 코드 추가 등에 중점을 둔다[13].

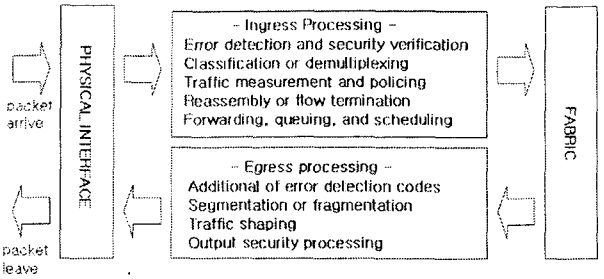

그림 2. ingress/egress 프로세싱과 data flow

[그림 2]는 ingress/egress 프로세싱의 개념 위주로 그린 것으로, 이와 같이 하는 일이 비슷한 부분을 묶어 네트워크 프로세서를 디자인하면 네트워크 프로세서 특유의 디자인 복잡성을 줄일 수 있다.

3.2 아키텍처에 따른 패킷 흐름

[그림 3]은 트래픽 관리를 위한 네트워크 프로세서의 전형적인 아키텍처 모습이다.

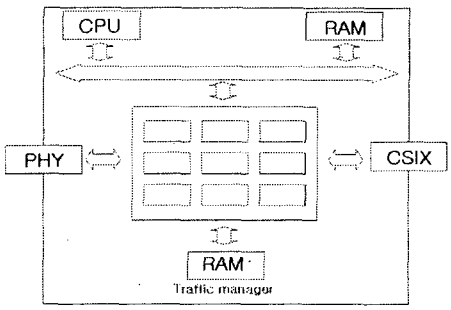

그림 3. 전형적인 네트워크 프로세서 아키텍처

[그림 3]과 같은 종류의 네트워크 프로세서는 임베디드 아키텍처 프로세서, 병렬 아키텍처 프로세서, 파이프라인 아키텍처 프로세서, 데이터플로우의 카테고리로 나눌 수 있으며, 가장 큰 차이점은 traffic flow이다.

먼저 임베디드 프로세서는 패킷마다 작동을 한다. 예를 들면 f, g, h 세 가지 실행을 순차적으로 한다고 가정하면 임베디드 프로세서는 각 패킷마다 f, g, h의 task를 실행한다[그림 4][13].

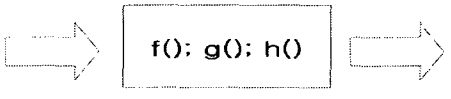

그림 4. embedded processor를 사용하는 아키텍처의 packet flow

병렬 아키텍처 프로세서는 들어오는 트래픽을 여러 개의 프로세서에서 나누어 처리한다. 만약, 프로세서가 N개라면 각 프로세서는 약 1/N의 트래픽을 처리한다[그림 5].

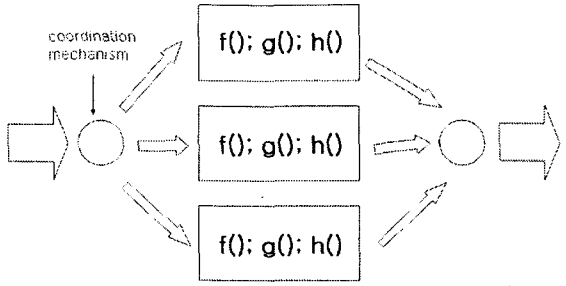

그림 5. parallel 아키텍처를 통한 packet flow

파이프라인 프로세서는 각각의 프로세서가 할당된 고유의 task를 수행한다[그림 6]. [13]

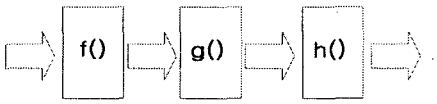

그림 6. pipeline 아키텍처를 통한 packet flow

데이터플로우 아키텍처는 대용량의 메모리로 구성되어야 하며, 메모리 안의 각 아이템은 태그를 갖고 있다. 이 태그는 필요한 프로세싱에 대한 상세한 정보를 포함하고 있다. 예를 들면, 어떤 프레임이 들어왔을 때 그 태그를 보고 어떤 그룹에 속하고 있는지 확인하고 프레임에 맞는 그룹으로 프레임을 보내고, 패킷을 받은 그룹은 그 패킷이 원하는 처리를 한다[13].

3.3 프로세서 계층의 기능 할당

네트워크 프로세서의 계층 구현은 특히 중요하다. 높은 성능을 위해서는 가장 낮은 레벨에 고속으로 대량의 패킷을 처리할 수 있도록 프로세싱을 할당해야 한다. 프로세서의 낮은 레벨 계층에서는 많은 데이터를 처리하기 때문에 bottleneck 현상이 발생할 수 있으며, bottleneck을 피하기 위해 data transfer, fabric interface, coprocessor hardware가 wire speed로 동작하게 해야 한다. [그림 7]은 프로세서 계층에 따른 패킷 흐름을 그림으로 나타낸 것이다[13].

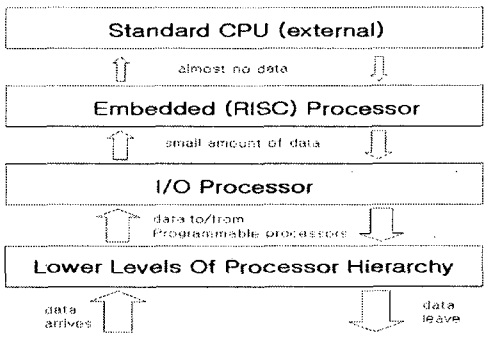

그림 7. processor hierarchy를 통한 packet flow

4. 네트워크 프로세서 비교

네트워크 프로세서 선택 기준은 상황에 따라 다르다. 어떤 곳에 쓸지, 어느 OSI 계층을 기준으로 어느 계층에 특히 중점을 두어야 하는지, 비용과 처리 속도 사이에서 어디에 중점을 둘지에 따라 채택되어지는 프로세서는 다양하다. 네트워크 프로세서 디자인을 하는 대표적인 회사는 Intel, IBM, Motorola가 있는데, 이번 섹션에서는 여러 종류의 네트워크 프로세서를 조사하여 네트워크 프로세서가 어디에 위치하는지와 그들의 내부 아키텍처의 구성을 2가지 관점에서 나누어 보았다.

4.1 Motorola C-5 Network Processor

C5NP는 셀이나 패킷 프로세싱, 라우터를 안정적으로 만들기 위한 큐 관리 기능을 가지며, 16 채널 프로세서(CP)와 table lookup, 큐 관리, 버퍼 관리, 스위칭 인터페이스와 control processor를 지원하는 5개의 장치를 가진 멀티프로세서 칩이다. 이러한 프로세서들을 컨트롤하는 프로세서를 제외하면 C5NP는 ASIC 타입과 유사하다. 각각의 CP는 내부 컨트롤을 위한 RISC core processor와 전송과 입력을 위한 serial data processors(SDP)들로 구성되어 있다.

C5NP는 주로 라우터를 위해 만들어졌으며, 2계층에서 7계층까지 처리할 수 있다. C5NP는 프로그래밍이 간단한 장점을 가지고 있다[10].

[그림 8]은 C5NP의 아키텍처 개요를 나타냈으며, [그림 9]는 C5NP의 파라미터에 대해 나타냈다.

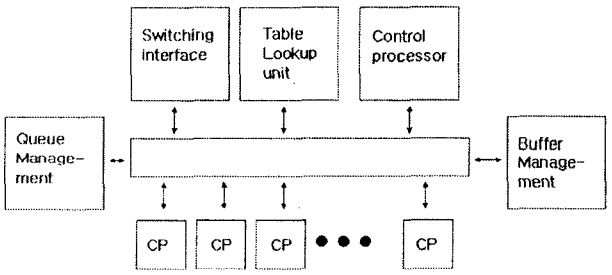

그림 8. C5NP 아키텍처 개요

CPU Performance (Max) (MIPS)	Operating Frequency (Max) (MHz)	Power Dissipation (Typ) (W)	Power Dissipation (Max) (W)	Throughput (Gbps)	Core Operating Voltage (Spec) (V)	I/O Operating Voltage (Max) (V)	Junction Temp (Min) (oC)
3400, 3961	200, 233	17.5, 20	20.5, 23	5	1.7	3.3	-40

Junction Temp (Max) (oC)	Integrated Memory Controller	Bus Interface	Fabric Interface	Networking Application Function	Manufacturing Process
100	SDRAM, SRAM	PCI	CSIX-L0, IBM PowerPRS	Data Plane	0.18

그림 9. c-5 파라미터

4.2 Intel IXP 1200 Network Processor

IXP 1200은 C5NP와 유사하게 동작하나 더 많은 구성 unit을 필요로 한다. IXP 1200은 166/200/232MHz의 SRAM과 6개의 microengine들로 구성되어 있다. Intel사의 특수한 microcode로 작동하

는 microengine이 모든 패킷 프로세싱을 수행하며, 이는 serial data processor(SDP)와 비슷한 기능을 수행한다. IXP 1200은 초당 2.5Mpacket을 처리할 수 있다. PCI 인터페이스에 외부 프로세서를 연결함으로써 높은 레이어를 지원할 수 있다[8].

4.3 Agere의 FPP, RSP, ASI

Fast Pattern Processor(FPP), Routing Switch Processor(RSP), Agere System Interface(ASI) 3개 칩 솔루션은 큰 라우터의 line card를 위해 만들어졌다. 세 개의 칩은 classification, policing, traffic management, QoS, traffic shaping, packet modification 기능을 수행한다. FPP와 RSP는 라우터의 fast path에서 테스크를 수행하며, ASI는 fast path와 slow path 둘 다 테스크를 수행한다. FPP와 RSP는 2.5Gbps wire rate로 프로토콜 분류를 한다.

내부적으로 FPP가 여러 개의 block을 만들고 병렬로 처리한다. input framer가 데이터 스트림을 64 byte로 나누고, 그 외의 프로세서들은 이 블록을 처리한다. RSP는 내부적으로 Application Specific Instruction Processor(ASIP)인 3개의 VLIW(very long instruction word processor)를 만들고, 이들 각각은 traffic management, traffic shaping, stream editing을 전담한다. ASI는 configuration, microprocessor bus로 interfacing, FPP의 지원을 위한 ASIC이다[7].

4.4 AMCC nP7120 (GPIF-200)

nP7120은 대형 라우터의 line card를 위한 네트워크 프로세서이다. 이는 패킷 classification과 modification을 포함하여 패킷을 처리한다. nP7120은 네트워크 프로세싱을 최적화한 2개의 코어 프로세서로 구성되어 있다[14].

4.5 Ezchip NP-1

NP-1은 대형 라우터의 line card를 위한 네트워크 프로세서이다. NP-1은 layer 7까지 프로토콜 프로세싱을 수행하므로, flow-based traffic policing과 URL switching에 사용될 수 있다. NP-1은 프로세서 코어들의 배열로 구성된다. 각 프로세서는 parsing, searching, resolving, modification을 위한 특정한 작업에 최적화되어 있다[2].

4.6 IBM PowerNP 4GS3

PowerNP는 single chip router같은 독립적인 기능을 하거나 더 큰 라우터의 일부분으로 될 수 있다. 2개의 PowerNP들이 중간 사이즈의 라우터에 직접적으로 연결되어 있거나, 큰 라우터의 line card에 위치할 수 있다. PowerNP는 데이터 처리를 할 때 fast path processing과 slow path processing 두 가지로 처리할 수 있다.

내부적으로 고정된 기능의 MAC와 큐 매니지먼트와 스케쥴링 unit으로 둘러싸여진 Embedded Processor Complex(EPC)로 구성되어 있다[8][15].

4.7 Clearwater Networks CNP810SP

CNP810SP는 slow path를 처리하는 프로세싱 부분을 제거하기 위해 line card에 쓰이는 네트워크 서비스 프로세서이다. CNP810SP는 MIPS 코드를 실행하는 simultaneous multi-thread core로 구성되어 있으며, 8개 명령 큐는 레스터의 분산된 세트와 함께 10개 기능 유닛을 병렬적으로 실행시킬 수 있다. CNP810SP는 약간의 instruction을 추가해서 기능을 확장할 수 있다. 이른바 네트워크 프로세싱 확장이다. 프로그래밍 면에서 보면, 최적화한 instruction된 set의 RISC에 속하나 아키텍처를 보면 전통적인 RISC core와 다른 면이 있다[12].

4.8 Lantronix DSTni LX, EX

Lantronix DSTni LX와 EX 칩은 높은 성능과 낮은 가격이 장점이다. 이 칩은 작은 임베디드 애플리케이션이나 이더넷 어플리케이션을 위주로 사용된다.

4.9 비교

네트워크 프로세서 등장 배경과 네트워크 프로세서 아키텍처에 대해 요약하여 살펴보면 다음과 같다.

먼저 네트워크 속도가 증가하고, 광섬유의 bandwidth가 점차 커짐에 따라 데이터의 전달 속도 또한 매우 빨라졌다. 2세대 네트워크 시스템이 등장했을 무렵 Data Bit Rate는 CPU clock 속도보다 더 빨라지게 되었다. 이 speed를 보조하기 위해서 네트워크 프로세서의 발전이 계속되었다. 한편으로 데이터 송수신에만 중점을 두어 프로세싱한 것에 비해 인터넷이 발달함에 따라 개개의 패킷 프로세싱이 해야 할 일들이 많아진 것도 네트워크 프로세서가 발전한 이유 중 하나이다.

네트워크 프로세서는 짧은 시간에 많은 패킷들을 처리해야 하며, 이외에도 네트워크 프로세서가 더 빨리 동작하기 위해 classification, modification, queuing, buffer management 등을 wire speed로 수행해야 한다.

다양한 네트워크 프로세서는 multiprocessor 아키텍처와 coprocessor 아키텍처로 분류될 수 있다. coprocessor는 여러 가지 방법으로 트래픽 관리를 하는데, 이는 임베디드 프로세서를 사용한 아키텍처, 파이프라인을 사용한 아키텍처, parallel을 사용한 아키텍처로 분류할 수 있다.

널리 쓰이는 네트워크 프로세서들을 조사한 바에 의하면 네트워크 프로세서들은 대부분 라우터와 line card에 쓰이며, 적용되는 응용 계층들은 다양하였다. IBM에서 나온 프로세서는 주로 라우터에, Agere에서 나온 프로세서는 주로 Line Card에 반영됨을 알 수 있었다. 프로세서들을 응용할 때 모토로라사가 제일 광범위하게 OSI 7계층 중 대부분을 응용할 수 있다.

아키텍처는 하나의 아키텍처를 쓰기보다는 각각의 아키텍처를 서로 보완하여 쓰인다. 보통 ASIC 보다 저렴하고 프로그래밍하기 쉬운 RISC 위주였으며, RISC와 함께 RISC의 단점을 보완하기 위해 ASIP이나 ASIC을 같이 사용하는 것으로 나타났다.

[표 1]은 앞에서 설명한 네트워크 프로세서 아키텍처들을 정리한 표이다.

표 1. 네트워크 프로세서 분류

processor	system aspects	application	architecture
Motorola C-5	Router	layer 2-7	RISC+ASIP
Intel IXP 1200	Router	layer 2-4	RISC+ASIP
Agere FPP,RSP, ASI	Line card	data path	RISC+ASIC
Agres NP 10 and TM 10	Line card	data path	ASIP+ASIC
IBM PowerNP 4GS3	Router	layer 2-4	RISC
AMCC nP7120	Line Card	layer 2-4	RISC+ASIC
AMCC nP3400	Router	layer 3-7	RISC
Broadcom BCM 5632	Router	layer 2	ASIC
Clearwater Networks CNP810SP	Router	control path	RISC
Lantronix DSTni	Terminal	layer2-4	general-purpose CPU+ASIC
KTH Protocol Processor	Router	IP forwarding	RISC

5. 결 론

네트워크 프로세서를 사용하여 네트워크 시스템을 개발할 때 네트워크 프로세서를 선택할 때는 어떤 목적 위주로 선택할지 신중히 생각해야 한다. 본 논문에서는 네트워크 프로세서를 선택할 때 도움이 되고자 알려진 네트워크 프로세서 중 일부분을 소개하였다.

본 논문에서는 네트워크 프로세서들이 어디서 동작하는지, 어떤 아키텍처를 썼는지 분류하였다. 네트워크 프로세서는 복잡하며 디자인 또한 다양하다. 대부분 네트워크 프로세서는 프로토콜 프로세서 영역에서 문제를 해결하기 위해 발전하고 있다. 어떤 아키텍처는 적은 instruction 수로 빠르게 동작하기 위한 목적이 있을 수 있고, 어떤 아키텍처는 full programmability를 지원해 가능한 모든 프로세싱을 처리하기 위해 만들어졌다.

네트워크 프로세서는 빠른 속도로 데이터들을 처리하며, 프로그래밍하기 쉬운 소프트웨어 아키텍처 위주로 발전하고 있으며 다양한 제품들이 나오고 있다. 본 논문에서 보듯이 네트워크 프로세서는 router나 line card 등의 다양한 위치에 위치하며, OSI 계층의 사용범위가 다양함을 보여주었다.

[참 고 문 헌]

[1] http://www.cs.ucla.edu/~rohitk/NetworkProcessor.htm, Rohit Kapoor, "Network Processors"
[2] http://www.ezchip.com/html/in_prod.html
[3] http://www.inf.fu-berlin.de/lehre/WS94/RA/RISC-9.html, "RISC concept-A Survey of Implementations"
[4] http://shekel.jct.ac.il/~citron/ca/isa.html, "Instruction Set Architecture(ISA)"
[5] futuresoft, "Challenges in Building Network Processor Based Solutions", White Paper, 2002
[6] http://www.agere.com/enterprise_metro_access/network_processors.html
[7] http://nps.agere.com/support/non-nda/
[8] http://developer.intel.com/design/network/products/family/ixp1300.htm
[9] http://www.embedded.com/story/OEG20010730S0053
[10] http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=C-5&nodeId=01DFTQ3126q62S
[11] http://tech.nplogic.com/
[12] http://www.zytek.com/~melvin/clearwater.html
[13] Douglas E. Comer, "Network System Design using Network Processors",
[14] https://www.amcc.com/cardiff/docManagement/displayProductSummary.jsp?prodId=nP7120
[15] http://www.intel.com/design/network/products/npfamily/ixp1200.htm
[16] Xiaoning Nie and Lajos Gazis , Framk Engle and Gerhard Fettweis, "A New Network Processor Architecture for High-speed Communication", 1999
[17] Niraj Shah, Kurt Keutzer, "Network Processor: Origin of Species", 2002