

Verilog-HDL을 이용한 다중필드 패킷분류 알고리즘의 설계 검증

The verification of the hardware implementation of packet classification algorithm on multiple fields by Verilog-HDL

홍성표*, 김준형**, 최원호***

* 경북대학교 전자공학과(전화:0(53)940-8857, 팩스:(053)950-5505, E-mail : oldtree@dgb.co.kr)

** 경북대학교 전자공학과(전화:(053)940-8857, 팩스:(053)950-5505, E-mail : boyjun95e@lycos.co.kr)

*** 경북대학교 전자공학과(전화:0(53)940-8857, 팩스:(053)950-5505, E-mail : @kr_prince@hotmail.com)

Abstract : This paper reports the RFC(Recursive Flow Classification) algorithm that is available on multiple fields. It is easy to be implemented by both software and hardware. For high speed classification of packets, the implementation of RFC is essential by hardware. Hence, in this paper, RFC algorithm is simulated by Verilog-HDL, and it verify the efficiency of the algorithm. The result shows that the algorithm can perform a packet classification within several cycles. It is not only much faster than software implementation but also enough to support OC192c.

Keywords : Verilog, HDL, Packet, Classification, RFC, multiple fields

1. 서론

최근까지의 인터넷 서비스는 단순히 먼저 도착한 Packet 을 먼저 서비스하는 최선형 서비스(Best-effort service)만 제공해 왔다. 하지만 인터넷 사용자의 폭발적인 증가로 인해 많은 양의 트래픽(Traffic)이 발생하게 되었고 사용자들은 서비스 품질 보장과 서비스의 차별화를 요구하게 되었다. 다시 말해 어떤 패킷에 대해 그 패킷을 보낼 것(Forwarding)인지 거를 것(Filtering)인지, 어디로 보낼 것(Routing)인지, 어떤 등급의 서비스를 부여할 것(Classification)인지, 또는 얼마를 과금할 것(Accounting & Billing)인지를 결정할 필요가 생긴 것이다. 패킷의 내용을 기초로 하여 패킷을 구분하는 일종의 규칙들을 만들고 그 규칙에 따라 패킷을 분류함으로써 다른 서비스를 제공할 수 있다. 이것을 패킷분류(Packetclassification)라 한다. 그림 1은 패킷분류가 인터넷 서비스 제공자(ISP)에 의해 어떻게 사용되어지는가를 보여준다.[3]

ISP1은 기업 네트워크 E1과 E2에그리고 NAP(Network Access Point)에 연결되어 있으며 NAP는 교대로 ISP2와 ISP1, ISP3에 연결되어 고객들에게 차별화된 서비스를 제공한다. 표1에서는 그 서비스들의 예를 보이고 있다.[3]

표 1. 패킷분류에 의한 차별화된 서비스의 예
Table 1. Examples of packet classification

Service	Example
Packet Filtering	경로 X를 통한 ISP1에서 E2로 가는 모든 트래픽을 거부하라.
Policy routing	경로 Y를 통해 E1으로부터의 모든 Voice-Over-IP 트래픽을 보내고 E2로의 트래픽은 ATM을 경유하여 보내라.
Accounting & Billing	경로 Y를 통해 E1으로 가는 모든 Video 트래픽에 대해 가장 우선 순위를 부여하고 그 트래픽에 대해 과금하라.
Traffic Rate Limiting	ISP2로는 10Mbps 이상의 E-mail 트래픽이 걸리지 않도록 하고 전체 트래픽이 경로 X에서 50Mbps를 넘지 않게 하라.
Traffic Shaping	경로 X를 통한 ISP2로의 Web 트래픽이 50Mbps를 넘지 않게 하라.

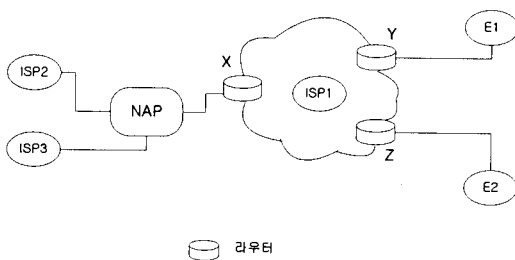


그림 1. 인터넷 서비스 제공자와 사용자 사이의 네트워크 구성도

Fig 1. Example network between ISPs and customers

패킷분류는 그 효율성이 점점 높아질 것이며 그에 따라 더 복잡한 방식과 룰(Rule)들이 생겨날 것이다. 이에 따라 패킷분류의 효율성을 높이기 위해 여러 알고리즘들이 연구되어져 왔다. 그 알고리즘들을 살펴보면, Basic Data Structure 형태의 Linear Searching, Caching, Hierarchical Tries, Set-Pruning Tries가 있고 Geometry 기반의 Grid-Of-Tries, Area-Based Quadtree, Fat Inverted Segment Tree 등이 있다, 그리고 Heuristic한 Recursive Flow Classification), Hierarchical Cuttings, Tuple-Space Search가 있다. 그 외에도 하드웨어적 기반의 Ternary CAM을 이용한 응용과 Bitmap-Intersection이 대표적인 연구들이라 하겠다.[1][4][5][10] 그 중 다중필드(Multiple Field)를 지원하고, 비연속 마스크(Non-Continuous Mask)를 지원하는 알고리즘은 RFC(Recursive Flow Classification)이다. 이 알고리즘은 소프트웨어와 하드웨어 두 방식 모두를 가지고 구현하기에 용이하며 소프트웨어로 구현했을 경우 OC48c(2.5Gb/s), 하드웨어로 구현했을 경우 OC192c(10Gb/s)를 지원할 수 있다.[3]

보다 빠른 패킷처리 속도(OC192c 이상)를 지원하기 위해 이 알고리즘은 하드웨어로 구현되어질 필요성이 있으며 본 논문은 RFC 알고리즘을 하드웨어로 구현하기 전 Verilog-HDL을 이용하여 설계·구현하고 그 성능을 검증하였다.

II. RFC 알고리즘

1. RFC 알고리즘의 구조

패킷을 분류한다는 것은 패킷 안의 S-bit를 T-bit의 클래스 아이디(Class Identifier or Class ID)로 매핑(Mapping)하는 것이라 볼 수 있다. RFC는 여러 단계를 거쳐서 원하는 값인 클래스 아이디를 얻는 알고리즘이다. 여기서 각 단계는 페이지(Phase)라 하며, 페이지의 수는 RFC 알고리즘의 성능과 특성을 결정한다.

RFC 알고리즘은 그림3에서처럼 여러 개의 단계로 나누어 메모리 낭비를 줄이고자 한다. 여러 단계를 거치기 위해선 S-bit의 룰을 효율적으로 나누어 그 정보를 메모리에 저장하며 각 단계마다 축소(Reduction)을 통해 메모리 사용의 효율을 높인다. RFC는 각 단계를 페이지(Phase)라 부르고, 그 페이지 수에 따라 RFC의 성능이 결정되어진다.[3]

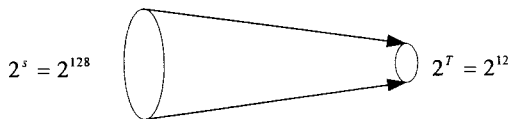


그림 2. 간단한 One-Step 분류
Fig 2. Simple One-Step Classification

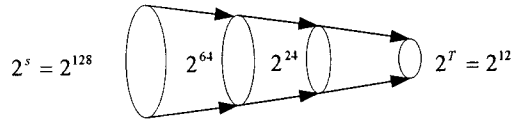


그림 3. RFC 구조
Fig 3. Structure of Recursive Flow Classification

2. RFC 알고리즘의 성능

패킷 분류 알고리즘의 성능은 일반적으로 룰(Rule)의 유연성, 처리할 수 있는 패킷 헤더 필드의 수, 저장 공간의 사용 정도, 업데이트 속도, 패킷의 처리 속도에 의해 결정된다.[1][2]

RFC의 성능은 두 가지의 파라미터(Parameter)로 결정된다. 하나는 페이지(Phase)의 수 P이고 다른 하나는 주어진 P에서 각각의 룰업 테이블을 어떻게 조합하여 어떤 형태의 축소 트리로 만들 것인가를 결정하는 것이다.

RFC 알고리즘에서는 페이지의 수를 증가시킬수록 메모리 소비는 줄어들지만 룰업 시간은 증가하게 되므로 그 사용 용도에 맞게 설계할 필요가 있다. 또한 같은 페이지 수를 사용하더라도 어떤 형태의 축소 트리를 사용하느냐에 따라 RFC 알고리즘의 성능은 달라지며 축소 트리 선택에 있어서 가장 크게 고려되어야 할 요소는 CES이다. 즉 중복되는 값이 많고 연관성이 깊은 Chunk를 묶어 CES를 최소화하는 것이 중요하다. 다음 그림 4, 5는 축소 트리의 선택 예를 보여 준다.[3]

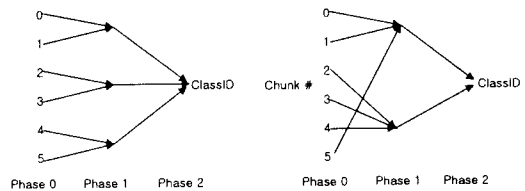


그림 4. RFC 3 페이지 축소 트리의 예
Fig 4. Two example reduction trees for three phases in RFC

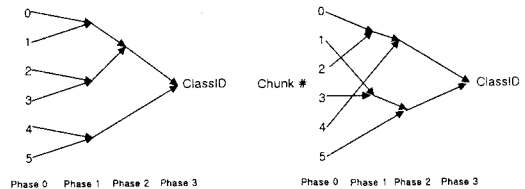


그림 5. RFC 4 페이지 축소 트리의 예
Fig 5. Two example reduction trees for four phases in RFC

3. RFC 알고리즘의 룩업 과정

패킷이 들어오면 첫 페이지에서 패킷 헤더를 분리하고 이렇게 분리된 값으로 각 Chunk를 룩업 한다. 룩업 결과는 다음 단을 위한 인덱스 계산에 사용되어 진다. 다음 단계에서는 앞단의 인덱스 계산 결과를 입력으로 받아 다시 룩업을 실행하고 그 결과 값을 사용하여 마지막 단계의 인덱스를 계산한다. 마지막 단계에서는 인덱스가 가리키는 룩업 테이블의 출력 값 클래스 아이디이므로 그 자체가 룩업 번호를 가리키게 된다. 표 2는 4개의 필드(Field)로 구성된 4개의 룰(Rule)로 구성되어 있다.

표 2. 룰 예
Table 2. An example of rule

Rule #	Source Add/Mask	Destination Add/Mask	Prot.	Dest. Port
0	0.76.140.5 /0.0.0.0	17.2.17.2 /0.0.0.0	udp	21
1	0.76.0.0 /0.0.0.0	17.2.0.0 /0.0.255.255	*	21
2	0.0.0.0 /255.255.255.255	17.2.0.0 /0.0.255.255	udp	*
3	0.76.0.0 /0.0.255.255	0.0.0.0 /255.255.255.255	*	*

그림 6은 표 2의 룰 예를 이용하여 RFC의 룩업과정을 보여준다.

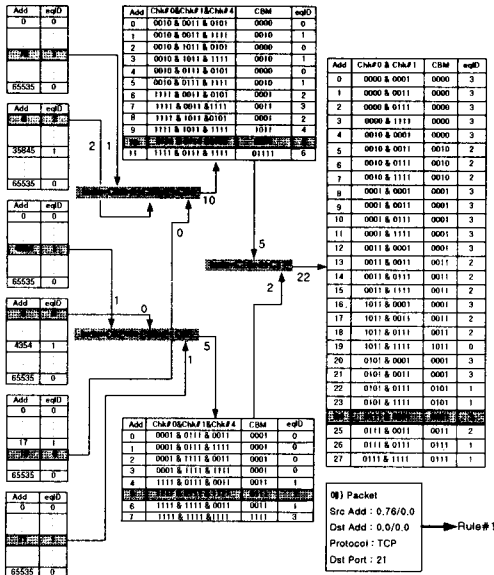


그림 6. RFC 룩업 과정

Fig 6. The Lookup Processing of RFC

III. 구현 결과에 대한 고찰

본 논문에서 구현한 패킷 분류기는 4 필드 5개의 룰을 가지며, 가장 효율적인 4 페이지의 축소 트리를 선택하여 RFC 알고리즘을 구현하였다. 그림 7은 각 단계의 입력과 출력의 흐름을 나타낸다.

- Stage 1 : RFC Table 0의 룩업과 Indx 1, Indx 2의 계산
- Stage 2 : RFC Table 1의 룩업과 Indx 3의 계산
- Stage 3 : RFC Table 1의 룩업 후 Rule# 도출

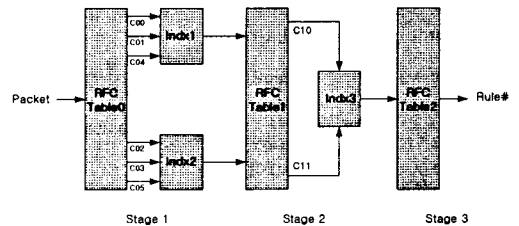


그림 7 RFC 각 단계에서의 입력 및 출력의 흐름

Fig 7. The flow of inputs & outputs in each stage of RFC 6-cycle

아래 그림 8은 각 단계에서 계산된 인덱스 값(c00, c01, c02, c03, c04, c05, c10, c11)과 매칭룰의 결과 값(Rule#)을 보여 준다. 이 그림에서 6-Cycle 동안 패킷이 들어와 룩업 과정과 인덱스 계산과정을 완료함을 볼 수 있다. 다시 말해 180Mhz 이상의 클럭 속도에서 초당 약 30x106 개의 패킷을 처리할 수 있으며 OC192c(10Gb/s)를 지원할 수 있다. 물론 높은 클럭 속도나 앞에서 언급한 파이프 라인으로 하드웨어를 설계했을 경우 그 보다 더 좋은 성능을 구현할 수 있다.

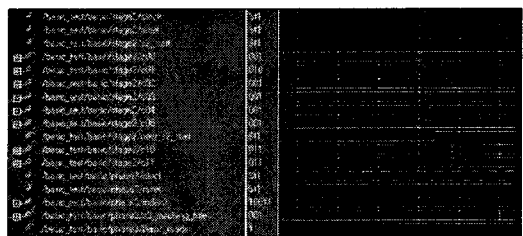


그림 8. RFC의 시뮬레이션 결과

Fig 8. The result of a simulation of RFC

그림 9는 합성틀인 Xilinx Foundation에서의 합성 결과를 나타낸다

Xilinx Mapping Report File for Design 'Lookup_top'
 Copyright (c) 1995-2000 Xilinx, Inc. All rights reserved.

Design Information

Command Line : map -p xcv600e-8-hq240 -o map.ncd
 hong1.ngd hong1.pcf
 Target Device : xv600e
 Target Package : hq240
 Target Speed : -8
 Mapper Version : virtexe -- D.19
 Mapped Date : Mon Oct 27 12:05:02 2003

Design Summary

Number of errors: 0
 Number of warnings: 38
 Number of Slices: 4,113 out of 6,912 59%
 Number of Slices containing
 unrelated logic: 0 out of 4,113 0%
 Total Number Slice Registers: 2,420 out of 13,824 17%
 Number used as Flip Flops: 2,385
 Number used as Latches: 35
 Number of 4 input LUTs: 6,194 out of 13,824 44%
 Number of bonded IOBs: 156 out of 158 98%
 Number of GCLKs: 4 out of 4 100%
 Number of GCLKIOBs: 1 out of 4 25%
 Total equivalent gate count for design: 56,581
 Additional JTAG gate count for IOBs: 7,536

그림 9. 구현된 RFC의 합성결과
 Fig. 9. The synthesis result of RFC

IV. 결론

최근 인터넷의 수요가 폭발적으로 증가함에 따라 대역폭의 한계를 가져왔고 이에 사용자들은 다양하고 차별화된 품질의 서비스를 요구하게 되었다. 따라서 패킷 분류기의 필요성이 커졌으며 또한 그 속도도 라우터와 더불어 높아져야 할 필요성이 있다. 하지만 다중필드(Multiple Field)를 지원하면서 높은 분류 속도와 메모리 소비의 효율을 높이는 것은 어렵다. 이러한 요구에 부응하는 알고리즘 중 RFC(Recursive Flow Classification) 알고리즘은 다중필드(Multiple Field)를 지원하며 높은 패킷 분류 속도와 메모리 소비의 효율성을 제공한다. 이 알고리즘은 소프트웨어와 하드웨어 두 방식 모두를 가지고 구현하기에 용이하며 소프트웨어로 구현했을 경우 OC48c(2.5Gb/s), 하드웨어로 구현했을 경우 OC192c(10Gb/s)를 지원할 수 있다. 보다 빠른 패킷처리 속도(OC192c 이상)를 지원하기 위해 이 알고리즘을 하드웨어로 구현할 필요성이 있으며 본 논문은 RFC 알고리즘을 하드웨어로 구현하기 전 Verilog-HDL을 이용하여 설계·구현하고 그 성능을 검증하였다.

참고문헌

[1] Pankaj Gupta and Nick McKeown, Stanford University, "Algorithms for Packet Classification", IEEE Network, March/April 2001.

[2] P. Gupta, S. Lin, and N. McKeown, "Routing lookups in hardware at memory access speeds," in Proceedings of the Conference on Computer Communications (IEEE INFOCOMM), (San Francisco, California), vol. 3, pp.1241-1248, March/April 1998.

[3] P. Gupta and N. McKeown, "Packet Classification on Multiple Fields", Proc. Sigcomm, Comp.l Commun. Rev. Vol.3, 2000, pp.1193~1202.

[4] J. van Lunteren and A.P.J Engbersen, "Multi-field packet classification using ternary CAM", ELECTRONICS LETTERS 3rd January 2002 vol. 38 No 1. pp.21~23.

[5] T.V Lakshman and D. Stiliadis, "High-Speed Policy-based Packet Forwarding Using Efficient Multi-dimensional Range Matching", Proc. ACM SIGCOMM, pp.191~202, September 1998.

[6] P.Gupta, S. Lin, and N. McKeown, "Routing lookups in hardware at memory access speeds," in Proceeding of the Conference on Computer Communications (IEEE INFOCOMM), (San Francisco, California), vol. 3, pp.1251~1260, April 1996.

[7] Steven McCanne and Van Jacobson, "A BSD Packet Filter: A New Architecture for User-level Packet Capture." in Proc. of Usenix Winter Conference, (San Diego, California), pp.259~269, Usenix, January 1993.

[8] Sally Floyd Van Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Transactions on Networking, vol. 1, pp.397~413. August 1993.

[9] P. Newman, T.Lyon and G.Minshall, "Flow labelled IP a connectionless approach to ATM", Proceedings of the Conference on Computer Communications (IEEE INFOCOMM), (San Francisco, California), vol. 3, pp.1251~1260, April 1996.

[10] V. Srinivasan, S. Suri, and G. Varghese, "Packet Classification using tuple Space Search" Proc. ACM Sigcomm, Sept. 1999, pp.135~46