TCAM을 이용한 패킷 분류를 위한 효율적인 갱신 알고리즘

준회원 정 해 진*, 송 일 섭*, 정회원 이 유 경**, 권 택 근*†

An Efficient Update Algorithm for Packet Classification With TCAM

Haejin Jeong*, Ilseop Song* *Associate Members*,
Yookyoung Lee**, Taeckgeun Kwon*† *Reguler Members*

요 약

고성능 라우터, 스위치 및 네트워크 보안 장비에서 패킷 포워딩 기능을 고속으로 수행하기 위해서는 효과적인 패킷 분류 기술이 필수적인데, 최근에는 TCAM과 검색엔진 등, 고속의 컨텐트 기반 검색 하드웨어를 이용하는 방법들이 사용되고 있다. 패킷 분류 시에는 트래픽 차단, 트래픽 모니터링 등의 목적을 위해서 많은 규칙들이 사용될 수 있고, 삽입과 삭제가 시스템 운용 중에 발생할 수 있다. 특히, 고속의 네트워크 환경에서 패킷 포워딩의 성능을 저하시키지 않기 위해서는 동적으로 변화하는 규칙들을 효과적으로 갱신하는 방법이 필요하다. 본 논문에서는 TCAM을 이용한 패킷 분류시 효과적인 갱신이나 재배치를 위해서 순서화된 부분 정렬 알고리즘을 제안하고, 실험을 통하여 TCAM의 이용률이 70%까지 높은 상황에서도 갱신으로 인한 재배치가 거의 일어나지 않도록 하여 재배치로 인한 패킷 처리의 지연을 줄일 수 있다는 결과를 보인다.

Key Words : TCAM, Packet Classification, Internet, Algorithm

ABSTRACT

Generally, it is essential that high-speed routers, switches, and network security appliances should have an efficient packet classification scheme in order to achieve the high-speed packet forwarding capability. For the multi-gigabit packet-processing network equipment the high-speed content search hardware such as TCAM and search engine is recently used to support the content-based packet inspection. During the packet classification process, hundreds and thousands of rules are applied to provide the network security policies regarding traffic screening, traffic monitoring, and traffic shaping. In addition, these rules could be dynamically changed during operations of systems if anomaly traffic patterns would vary. Particularly, in the high-speed network, an efficient algorithm that updates and reorganizes the packet classification rules is critical so as not to degrade the performance of the network device. In this paper, we have proposed an efficient update algorithm using a partial-ordering that can relocate the dynamically changing rules at the TCAM. Experimental results should that our algorithm does not need to relocate existing rules feature until 70% of TCAM utilization.

Ⅰ. 서 론

인터넷 트래픽의 수요를 만족시키기 위하여 네트워크 장비의 용량과 대역폭은 6개월마다 두 배씩 증가하고 있다[1]. 인터넷 백본망의 주요 스위치와 라우터는 2.5 Gbps, 10 Gbps, 또는 40 Gbps 의 속도를 지원하고 있으며, 사용자를 연결하는 액세스망의 인터페이스도 기가비트 이더넷 또는 10 Gbit

※ 본 연구는 정보통신부의 대학 ITRC 지원사업의 연구비 지원으로 수행되었음.
* 충남대학교 컴퓨터공학과, † 교신저자(tgkwon@cnu.ac.kr) ** 한국전자통신연구소
논문번호 : KICS2005-09-390, 접수일자 : 2005년 9월 29일

이더넷 환경으로 바뀌어가고 있다. 특히, 광대역 네트워크의 사용자의 인터넷 접속 속도도 xDSL 및 CATV의 1~8 Mbps 이내의 연결 속도에서 점차적으로 FTTH의 100 Mbps 이상의 기가비트 급으로 확산이 예상되고 있는 실정이다. 이러한 고속 인터넷 환경에서의 패킷 헤더의 목적지 주소를 테이블에서 찾아서 넥스트 홉(next hop)으로 전달하는 라우터에서 OC-192 같은 10 Gbps의 성능을 보장하는 것은 쉽지 않다. 왜냐하면, 라우터에서 서비스품질(QoS)과, 패킷 필터링과 같은 보안 기능 등을 처리하기 위해서는 기존의 단순한 패킷 포워딩과는 다른 복잡한 패킷 처리 과정을 수행해야 하기 때문에 고속의 라인 카드 전송률을 만족하기 어렵다. 패킷 분류는 들어온 패킷들을 플로우(flow)로 분류하는 것이다. 플로우는 IP/TCP/UDP 헤더의 출발지, 목적지 IP 주소, 출발지, 목적지 포트번호, 프로토콜의 필드들을 공유하는 패킷들의 흐름으로 정의된다. 패킷 분류는 일반적으로 라우팅 룩업 연산을 위해 필요하지만 패킷 필터링이나 일반적인 네트워크의 트래픽 관리 정책을 적용하기 위해서도 사용된다[2]. 따라서 고성능 패킷 분류와 고속의 라우팅 룩업에 관한 많은 연구들이 수행되고 있다. 특히, 멀티 기가비트 라인카드에서 고속의 하드웨어 예를 들어, TCAM(Ternary Content Addressable Memory), NP (Network Processor), ASIC 등을 이용하여 페이로드 검사 시의 오버헤드를 해결하기 위한 패킷 분류 기술과 라우팅 룩업을 위한 알고리즘에 관한 연구도 최근에 진행되고 있다. 예를 들어 [2, 5-7]의 연구에서는 고속 네트워크에서의 빠른 라우팅 룩업을 위한 패킷 분류 문제를 해결하는 알고리즘을 제안했고, [3, 4, 8-12] 등의 연구에서는 패킷 정책이나 동적인 규칙에 따르는 패킷 분류에 관한 문제를 해결하는 알고리즘을 개발하였다.

패킷 분류를 위해 필요한 규칙들은 새로운 규칙의 등장, 기존 규칙의 갱신 및 삭제 등의 이유로 인하여 동적으로 변할 수 있기 때문에, 고속의 패킷 포워딩을 수행하면서 성능 저하 없이 규칙들을 갱신할 수 있어야 한다. 그러나 기존 알고리즘들이 패킷 필터링을 위한 패킷 분류를 수행하면서 갱신이나 재배치로 인한 과부하로 수 기가 비트의 전송 속도를 유지하지 못한다[3]. 즉, 하나의 규칙을 추가하기 위하여 일정 시간 동안 서비스를 중단하지 않고, 규칙의 갱신이 수행되게 하는 알고리즘이 필요하다. 따라서 본 논문에서는 빠른 패킷 분류를 위해 룩업 시간이 빠른 TCAM과 고속의 네트워크 프로세스가 사용되는 환경에서 성능이 저하되지 않도록 하는 효율적인 패킷 분류 규칙의 갱신 및 재배치 알고리즘을 제안하고 성능을 분석하였다.

본 논문의 2장에서는 TCAM의 소개와 TCAM을 이용한 패킷 분류에 대하여 기술하고, 3장에서는 TCAM을 이용한 패킷 분류에서의 효율적인 갱신 및 재배치를 위한 알고리즘을 제시한다. 4장에서는 제안된 알고리즘의 실험 결과와 다른 갱신 알고리즘과의 비교 성능 평가를 하였다. 5장은 연구 결과와 향후 과제를 제시한다.

Ⅱ. TCAM을 이용한 패킷 분류

2.1 TCAM

TCAM은 고속 검색 기능을 제공하는 메모리이다. TCAM은 각 메모리들의 연계가 쉽고 TCAM의 각 셀마다 세 개의 로직 상태 중 하나의 상태를 유지한다. 세 개의 로직 상태는 '0', '1', 그리고 'don't care' 비트이다. 이 don't care 비트는 규칙의 어떠한 위치에도 올 수 있는 와일드 카드이고, 규칙의 패턴 매칭을 지원한다. 각 TCAM의 룩업은 1 사이클(cycle)만 필요하기 때문에 룩업 하는 시간이 매우 빠르다[3]. 이와 같은 이유로 TCAM은 멀티 기가비트의 성능을 위해 네트워크 프로세스에서 패킷 분류 작업에 많이 쓰이는 메모리이다. TCAM을 이용한 패킷 분류는 규칙을 저장하고 빠른 시간 안에 룩업하여 고속으로 유입된 패킷들에게 룰이 적용되도록 하는 일을 한다.

그림 1은 TCAM에서의 패턴 매칭의 예를 보여주고 있다. 위 그림과 같이 TCAM에 4 개의 엔트리(또는 룰)가 저장되어 있고, 위와 같이 [A G T G A T C G T]라는 텍스트가 들어 올 때 TCAM에서 룩업을 하게 되면 매칭 되는 두 개의 엔트리 3번째와 4번째 엔트리 중 첫 번째 매치되는 엔트리를 리턴한다. 즉, 들오온 패킷에 패턴 매칭 룰을 적용할 때 제일 먼저 일치하는 룰(그림에서 3번째 엔트리)이 적용 된다.

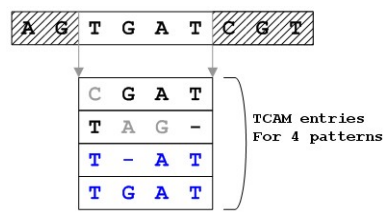


그림 1. TCAM의 패턴 매칭의 예

TCAM을 이용하여 패킷을 분류함으로써 고속의 링크, 예를 들어 OC-192의 초고속 인터넷 백본망에서 QoS 지원을 위한 패킷의 구분과 고속 방화벽 기능을 지원할 수 있다. OC-192의 링크에서는 하나의 패킷을 100 ns(nano-second) 이내에 처리해야 하므로 소프트웨어에서의 패킷 분류가 불가능하다.

2.2 기존의 TCAM을 이용한 패킷 분류

패킷 분류는 해당 패킷이 어떤 플로우에 속하는지 구분하는 것으로 IP 패킷의 경우에 5개의 필드를 가진 투플로 분류한다. 각 필드는 32 비트에서 8 비트로 표현되어 전체 플로우 테이블의 크기는 104 비트이다.

	32	32	16	16	8
Rule	srcIP	dstIP	srcPort	dstPort	prot
R5	10.0.0.0/8	11.0.0.0/8	any	any	any
R1	10.0.0.0/8	11.0.0.0/8	1024	80	tcp
R2	10.0.0.0/8	12.0.0.0/8	2048	21	tcp
R3	10.0.0.0/8	11.0.0.0/8	any	80	tcp
R4	10.0.0.0/8	any	1024	80	tcp

그림 2. TCAM을 이용한 패킷 분류를 위한 플로우 테이블

그림 2는 고속의 패킷 분류를 위하여 TCAM의 플로우 테이블을 통해 수행하는 경우의 예로써 4개의 플로우를 구분하는 규칙을 보여주고 있다. R1과 R2는 전체 5-투플의 모든 필드들을 이용하고 있고, R3와 R4의 경우는 각각 source port number(srcPort)와 destination IP address(dstIP)의 값을 “don’t care”로 사용하고 있다.

2.3 기존 TCAM을 이용한 패킷 분류의 문제점

TCAM을 이용한 플로우 테이블에서 테이블 엔트리, 즉, TCAM 엔트리의 순서는 중요하다. 예를 들어, 그림 2에서 R1과 R2의 순서가 바뀌는 경우에 입력되는 패킷의 srcPort와 무관하게 나머지 필드만 매치되는 경우에 R3에 매치되는 플로우의 패킷으로 분류한다. 하지만 R3와 R4의 경우와 같이 플로우를 기술하는 필드의 종류가 다르거나 R3 그리고 R4의 경우는 그 순서와 무관하게 플로우에 속하는 패킷을 분류할 수 있다.

본 논문에서는 이러한 점을 이용하여 모든 TCAM 엔트리의 순서를 유지하는 것보다 부분적인 TCAM 엔트리의 순서를 유지하여 TCAM 엔트리의 갱신시 TCAM 엔트리의 이동을 최소화함으로써 효율적인 TCAM 엔트리의 갱신이 되도록 한다.

R5 : 10.0.0.0/8, 11.0.0.0/8, any, any, any

(a) 삽입 전

Rule	srcIP	dstIP	srcPort	dstPort	prot
R1	10.0.0.0/8	11.0.0.0/8	1024	80	tcp
R2	10.0.0.0/8	12.0.0.0/8	2048	21	tcp
R3	10.0.0.0/8	11.0.0.0/8	any	80	tcp
R4	10.0.0.0/8	any	1024	80	tcp

(b) 삽입 후

Rule	srcIP	dstIP	srcPort	dstPort	prot
R5	10.0.0.0/8	11.0.0.0/8	any	any	any
R1	10.0.0.0/8	11.0.0.0/8	1024	80	tcp
R2	10.0.0.0/8	12.0.0.0/8	2048	21	tcp
R3	10.0.0.0/8	11.0.0.0/8	any	80	tcp
R4	10.0.0.0/8	any	1024	80	tcp

그림 3. TCAM에서의 룰의 우선 순에 따른 엔트리 이동 현황

하지만 패킷을 분류하는 규칙이 동적으로 변화하는 환경에서 TCAM 엔트리의 갱신 효율을 향상시킬 필요가 있다[1]. 예를 들어, 특정 서비스를 위해 하나의 룰을 추가해야 할 때 TCAM에서 재배치가 일어난다면 재배치를 하는 동안 서비스를 중단해야 하는 상황을 막아야 하는데, 이를 위하여 패킷 처리의 중단 없이 TCAM 엔트리의 갱신이 수행되거나 혹은 패킷 분류 시간 내에 TCAM 엔트리의 갱신이 수행되는 알고리즘이 필요하다.

Ⅲ. TCAM을 이용한 패킷 분류 갱신 알고리즘

3.1 순차적 정렬(Sequence ordering)

[2]의 논문에서 TCAM의 갱신 시간을 최소화하기 위하여 프리픽스의 길이별로 엔트리를 나누고 이들 엔트리 사이에 빈 공간을 두었다. 이를 본 논문에서도 활용하여 패킷 분류를 위한 필드 수에 따라 엔트리를 나누고 이들 엔트리 사이에 빈공간을 둘 경우에 새로운 엔트리가 저장될 공간을 쉽게 찾을 수 있다. 패킷 분류를 위하여 5 투플을 사용할 경우, TCAM 엔트리별 공간의 구성은 그림 4에서 보는 바와 같다. 그림 4와 같이 기본적으로 TCAM

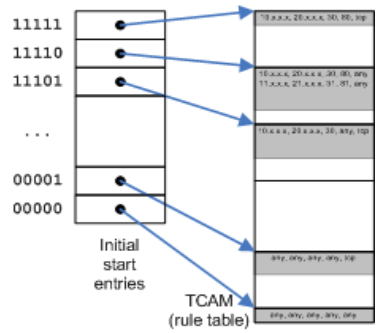


그림 4. 패킷 분류를 위한 TCAM 엔트리 순차 배치

의 각 그룹별로 엔트리 사이에 빈 공간을 두고 엔트리 추가 시 어느 한 그룹의 빈 공간이 없을 때까지 이 상태를 유지하도록 한다. 새로운 TCAM 엔트리의 추가는 각 투플 개수의 엔트리 사이 빈 공간에 저장되므로 기존 TCAM 엔트리의 재배치가 필요 없다.

하지만 빈 공간이 없어지면 새로운 엔트리의 추가를 위하여 인접한 투플 개수의 엔트리는 재배치되어야 한다. 최악의 경우 순차적 정렬에서는 엔트리 하나를 추가하기 위해 기존 엔트리들이 31번 이동해야 한다.

3.2 부분 정렬(Partial ordering)

새로운 패킷 분류 정책의 추가로 인하여 TCAM의 엔트리를 추가할 경우에 TCAM 엔트리의 순서가 중요하므로 추가되어야 하는 위치 이후의 모든 TCAM 엔트리를 이동하여야 한다. TCAM 엔트리의 수를 N이라 가정하면TCAM 갱신을 위한 알고리즘의 복잡도는 O(N)이 된다. 따라서 이러한 방법으로는 라우터 등 시스템의 운용 중에 TCAM 엔트리의 추가가 필요한 새로운 패킷 분류 정책의 변경이 어렵다.

TCAM 엔트리는 패킷 분류를 위하여 저장된 순서가 중요하지만 패킷 분류를 위한 필드가 다수 개 존재할 경우에는 전체 TCAM 엔트리의 순서가 아닌 해당 필드로 분류되는 TCAM 엔트리의 순서만 유지하면 된다. 따라서 서로 다른 필드로 분류되는 패킷 분류 정책은 서로 독립적인 관계를 가지므로 TCAM 엔트리의 순서를 유지할 필요가 없다.

그림 5은 편의상 4개의 필드, 즉, a, b, c, d 필드로 구성되는 분류 정책이 TCAM에 저장될 때의 종속 관계를 그래프로 표현한 것이다. 여기서 기호 '-'는 해당 필드가 "don't care"의 값을 가진 경우를 나타낸다.

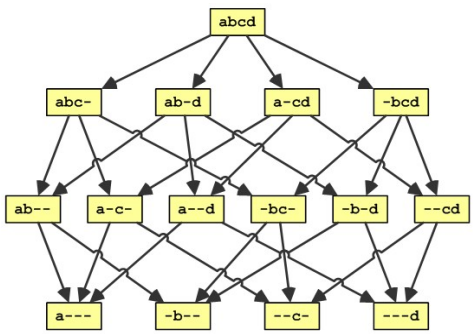


그림 5. 4개의 필드(a, b, c, d)로 구성된 TCAM 엔트리의 종속 관계 그래프

<a--->의 엔트리는 3개의 엔트리 <ab-->, <a-c->, <a--d>의 위치와 종속 관계에 있다. 따라서 <a--->는 종속 관계에 있는 3 개의 엔트리 이후에 나타나면 된다. 하지만 3 개의 엔트리 <ab-->, <a-c->, <a--d>의 각각은 서로 독립적인 관계가 있어 그 순서와 무관하다. 이러한 성질을 이용하여 TCAM 엔트리의 부분 정렬 관계를 설정할 수 있다. 예를 들어, <abcd>는 다른 모든 엔트리보다 앞서지만 <abc->, <ab-d>, <a-cd>, <-bcd>가 나머지 엔트리보다 앞서야 하므로 그림 5에서의 종속 관계 그래프가 성립한다. 이러한 종속 관계 그래프를 이용하면 새로운 엔트리의 추가는 부분 정렬의 조건을 만족하는 위치에 저장될 수 있다.

3.3 TCAM 재배치

TCAM 재배치는 자주 발생하진 않지만 한꺼번에 여러 TCAM 엔트리를 이동해야 하는 오버헤드가 발생한다. 이 경우에도 TCAM 엔트리의 부분 정렬에 필요한 엔트리 만을 이동함으로써 그 부담을 최소화 할 수 있다. 예를 들어, <ab-d>의 엔트리가 더 이상 추가될 빈 공간이 없을 경우에 <ab-d>에 종속 관계에 있는 <ab-->, <a--d>, <-b-d> 중 하나를 선택하여 재배치 한다. 이 경우에 각각의 엔트리에 종속 관계에 있는 <a--->, <-b-->, <---d> 엔트리 이전의 공간으로 이동하여야 연쇄적인 재배치가 일어나지 않는다. 그림 6은 재배치가 일어나는 종속 관계의 엔트리를 나타내고 있다.

예를 들어, <ab-d> 엔트리의 새로운 정책을 추가하기 위하여 <ab-->를 선택하였다고 가정하면 다음과 같다. <ab-->의 엔트리를 <a--->와 <-b-->의 최초 위치, 즉, first(A)를 <a---> 엔트리 그룹의 시작 주소 위치라고 하고 first(B)를 <-b--> 엔트리 그룹의 시작 주소 위치라고 하고 first(A)와 first(B) 사

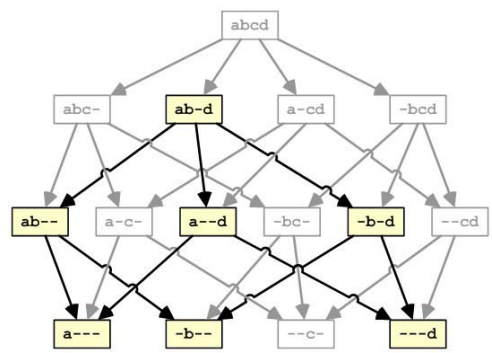


그림 6. TCAM 엔트리의 재배치를 위한 종속 관계 그래프

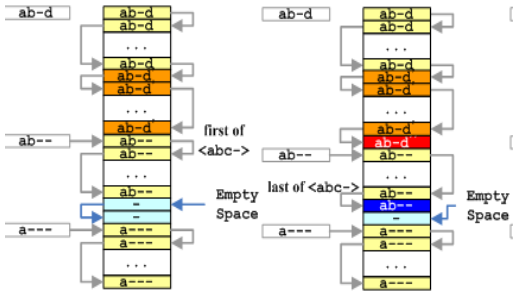


그림 7. TCAM 엔트리 추가에 따른 재배치: (a)<abcd> 엔트리가 추가되어 빈 공간을 모두 사용한 경우, (b) 새로운 <abcd> 엔트리 추가를 위하여 최초의 <abc-> 엔트리가 마지막으로 재배치된 경우

이의 공간을 interval(first(A), first(B)) 라고 한다면 interval(first(A), first(B)) 이전의 공간으로 이동함으로써 <ab-d> 엔트리의 새로운 정책을 추가하기 위한 빈 공간을 확보할 수 있다.

그림 7은 <ab-d> 엔트리 (ab-d')가 연속적으로 추가되어 <ab-d>와 종속 엔트리 사이의 빈 공간이 없어진 경우, 재배치가 일어나는 상황을 보이고 있다. 즉, 새로운 <ab-d> 엔트리 (ab-d'')의 추가를 위하여 <ab-->의 엔트리 중 첫 번째 엔트리 (ab--')를 선택하여 <ab--> 엔트리의 마지막 (ab--'')으로 이동함으로써 <ab-d> 삽입을 위한 새로운 공간이 생기고 여기에 <ab-d> 엔트리 (ab-d'')를 추가하여 부분 정렬을 유지한다. *M* 개의 투플을 패킷 분류에 사용할 경우에 TCAM 엔트리의 재배치는 O(*M*)이지만, 일반적으로 패킷 분류를 위하여 5-투플이 사용되므로 최악의 경우에 TCAM 엔트리의 재배치를 위한 이동횟수는 최대 4번만 발생한다. 다시 말해, don't care 비트를 사용한 수대로 그룹을 나눈다면 5개의 투플에서는 5개의 그룹이 생기는데 이 그룹별로 종속관계를 가진다면 한 개의 엔트리를 추가하기 위해서 그룹간의 종속관계를 검색한다. 따라서 총 4번의 종속관계 검색으로 인한 이동횟수가 발생한다. 이는 *N* 개의 엔트리를 가진 TCAM 엔트리를 트리의 형태로 재배치하는 경우의 복잡도가 엔트리 수에 의해 지수적으로 증가하는 O(*log*(*N*))인 것에 비하여 무시할 수 있을 정도의 작은 값이다.

Ⅳ. 성능 평가

TCAM 갱신 알고리즘의 성능을 비교 평가하기 위해, 각 투플 그룹 사이에 빈 공간을 동일한 크기로 나누어 가지지만 엔트리의 재배치를 순차적으로 하는 단순한 TCAM 갱신 알고리즘과 본 논문에서 제안한 종속 관계에 따라 재배치를 수행하는 부분 정렬 TCAM 갱신 알고리즘을 비교하여 실험하였다. 비교 대상은 하나의 엔트리를 추가하기 위해 TCAM내의 기존 엔트리의 이동 횟수와 TCAM에 저장될 수 있는 최대 엔트리 수에 따른 엔트리 이동 변화이다. 패킷 분류를 위한 규칙들은 무작위로 생성되도록 하였고, TCAM 엔트리의 삽입 요청이 있을 때 기존 TCAM에 저장된 엔트리들의 이동 횟수를 측정하였다. 이동 횟수의 측정은 알고리즘 내에서 TCAM의 엔트리가 이동을 위해 copy 되는 횟수를 카운팅 하는 프로그램을 넣어서 측정하였다.

성능 실험은 TCAM의 크기, 즉 저장 가능한 최대 엔트리 개수를 1024와 5120로 변화시켰고, 삽입 엔트리 개수는 한번에 10개의 엔트리를 10번씩 추가했다. 엔트리 이동 회수는 10번의 실험에 대한 평균 엔트리 이동 회수에 대해 측정하였다. 그림 8의 그래프는 10번의 실험 중 각각의 엔트리 추가 개수에 따른 이동 횟수의 최소값, 평균값, 및 최대값을 측정하였다. y축은 기존 엔트리들의 이동 횟수이고, x축은 TCAM에 저장된 엔트리의 수이다. 예를 들어, 그림 8의 (a)의 경우 x축의 100에서는 TCAM에 1000개의 엔트리가 있을 때 한 개의 엔트리를 삽입할 때 기존 엔트리의 이동 횟수의 최대값은 31이고, 평균값은 25, 최소값은 22이다.

그림 8의 그래프와 같이 두 개의 알고리즘은 엔트리의 이동수에 대해 현저한 차이를 보인다. 우선 순차적 정렬 TCAM 갱신 알고리즘에서는 TCAM 사이즈의 20% 용량이 차는 부분부터 이동수가 점차적으로 늘어가는 것을 볼 수 있고 이에 반해 순서화된 부분 정렬 TCAM 갱신 알고리즘에서는 70%부터 이동수가 증가하고 있다. 순차적 정렬 TCAM 갱신 알고리즘에서는 최대 31까지의 이동수를 보이고 순서화된 부분 정렬 TCAM 갱신 알고리즘에서는 최대 4까지의 이동수를 보여 무려 8배 정도의 성능 차이를 보이고 있다. 따라서 그림 8에서 보는 바와 같이 패킷 분류를 위한 저장 매체의 정책을 임의로 생성할 경우에 TCAM에서의 엔트리 이동 회수는 갱신 알고리즘의 성능을 결정하는 요인이다.

그림 9는 TCAM의 크기에 따른 순차적 정렬 TCAM 갱신 알고리즘과 순서화된 부분 정렬 TCAM 갱신 알고리즘에 대하여 10번의 엔트리 삽입시 요구되는 엔트리의 이동수에 대한 표준편차 그래프이다. 그래프의 y축은 TCAM에 저장된 엔트리 개수에 따른 표준편차이고, x축은 엔트리 삽입시

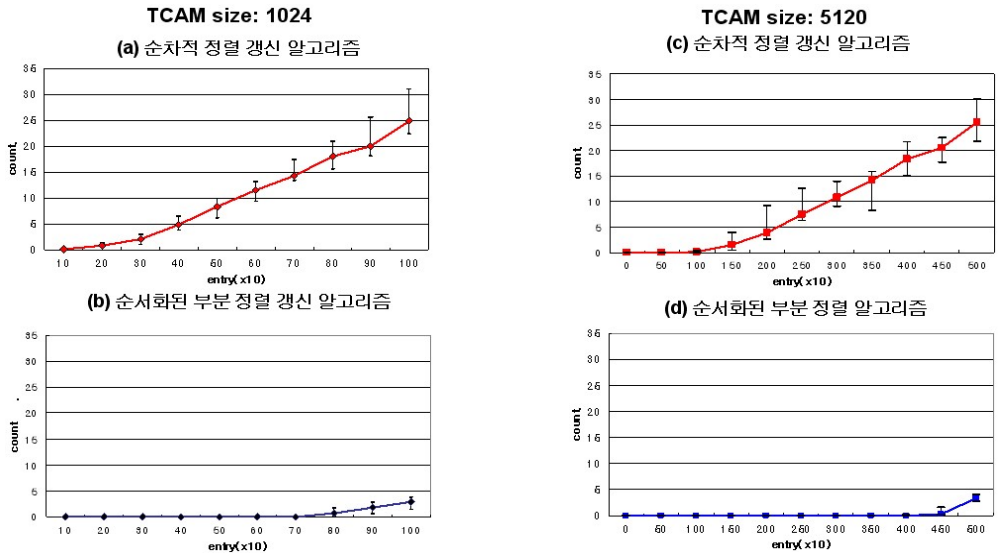


그림 8. 순차적 정렬 TCAM 갱신 알고리즘과 순서화된 부분 정렬 TCAM 갱신 알고리즘의 성능 비교 평가

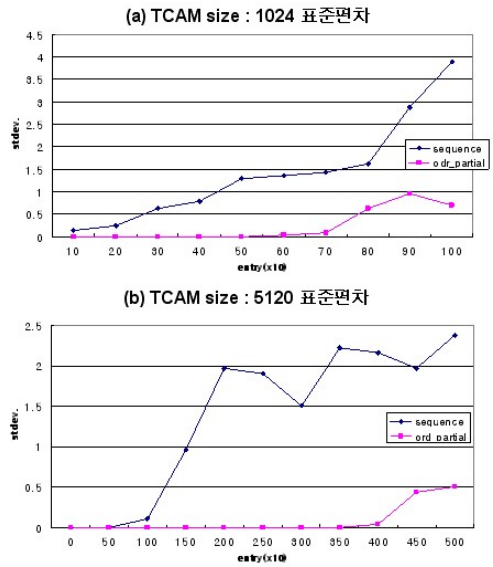


그림 9. 순차적 정렬 TCAM 갱신 알고리즘(a)과 순서화된 부분 정렬 TCAM 갱신 알고리즘(b)의 표준편차

TCAM에 저장된 엔트리 개수로 TCAM의 이용률을 나타낸다. 그림 9의 그래프에서 TCAM 사이즈가 1024 나 5120 일 때 모두 순서화된 부분 정렬 TCAM 갱신 알고리즘이 순차적 정렬 TCAM 갱신 알고리즘보다 이동수에 대한 표준편차가 작게 나타난다. 따라서 순서화된 부분 정렬 갱신 알고리즘이 순차적 정렬 갱신 알고리즘보다 빈 공간이 없을 시 엔트리 삽입 요청이 있을 때 TCAM의 재배치에 있어 더 결정적(Deterministic)임을 알 수 있다.

Ⅴ. 결 론

멀티 기가비트의 네트워크 장비에서 고성능의 패킷 분류를 수행하기 위해서는 고속의 네트워크 프로세서와 같은 하드웨어와 TCAM을 이용한 패킷 분류 알고리즘이 필수적인 기술로 인식되고 있다. TCAM 기반의 패킷 분류 기술은 TCAM에서의 갱신과 검색 시간이 성능 평가의 중요한 역할을 한다. 또한, 고속의 라우터에서는 패킷 필터링과 같은 기능을 업데이트하는 동적인 환경에서도 고속으로 수행할 수 있어야한다.

따라서 본 논문에서는 빠른 패킷 분류를 위해 룩업 시간이 빠른 TCAM과 고속의 네트워크 프로세스가 사용되는 환경에서 성능이 저하되지 않도록 하는 효율적인 패킷 분류 규칙의 갱신 및 재배치 알고리즘을 제안하고 성능을 분석하였다. 본 논문에서 제시한 부분 정렬 TCAM 갱신 알고리즘은 기존의 순차 정렬TCAM 갱신 알고리즘과 달리 TCAM 용량의 70% 정도까지는 갱신으로 인한 추가적인 재배치가 필요없다. 또한 많은 엔트리 수가 추가되어 TCAM 용량의 70%가 넘어 재배치가 자주 일어난다 하더라도 엔트리의 이동 횟수는 최악의 경우 최대 4번으로 줄였다. 결과적으로 부분 정렬 TCAM 갱신 알고리즘은 기존 TCAM 내용의 변경을 최소화하였다. 표준편차에 대한 실험을 통해 TCAM의 엔트리 이동수의 변화가 적고, 현재 처리 중인 패킷에 영향을 최소화하여 기존 TCAM의 갱신 부담이

상대적으로 비결정적이므로 고속 패킷 분류에 적용이 용이함을 보였다. 차후 라우터 개발 플랫폼인 Intel IXDP28XX에 IDT 75K62134 [13]의 9 Mbit TCAM을 장착해 구현하여 실제 패킷을 처리하면서 패킷 분류를 위한 정책을 임의로 생성할 경우에 대해 TCAM 갱신 알고리즘의 실제 성능을 평가할 것이다.

참 고 문 헌

〔1〕 F. Baboescu, S. Singh, and G. Varghese, “Packet Classification for Core Routers: Is There an Alternative to CAMs?,” IEEE INFOCOM, Mar. 2003.

〔2〕 D. Shahand P. Gupta, “Fast Incremental Updates on Ternary-CAMs for Routing Lookups and Packet Classification,” Proc. Hot Interconnects, Mar. 2000.

〔3〕 Z. Wang, H. Che, M. Kumar, and S. K. Das, “Consistent TCAM Policy Table Update with Zero Impact on Data Path Processing,” IEEE Computer, vol. 53, no. 12, pp. 1602-1614, Dec. 2004.

〔4〕 T. Lakshman and D. Stidialis. “High speed Policy-based Packet Forwarding using Efficient Multi-dimensional Range Matching,” Proc. ACM SIGCOMM, Sep. 1998.

〔5〕 P. Gupta, S. Lin, and N. McKeown, “Routing Lookup in Hardware at Memory Access Speeds,” Proc. INFORCOM, pp. 1240-1247, Mar. 1998.

〔6〕 A. Brodnik, S. Carlsson, M. Degermark, and S. Pink, “Small Forwarding Tables for Fast Routing Lookups,” Proc. ACM SIGCOMM, pp. 3-13, Oct. 1997.

〔7〕 M. Waldvogel, G Varghese, J. Turner, and B. Plattner, “Scalable High-speed IP Routing Lookups,” Proc. ACM SIGCOMM, pp. 25-36, Oct. 1997.

〔8〕 P. Gupta and N. McKeown, “Classifying Packets Using Hierarchical Intelligent Cuttings” IEEE Micro, vol. 20, no. 1, pp. 34-41, Jan-Feb 2000.

〔9〕 P. Gupta and N. McKeown, “Packet Classification on Multiple Fields,” Proc. ACM SIGCOMM, pp. 147-160, Sept. 1997.

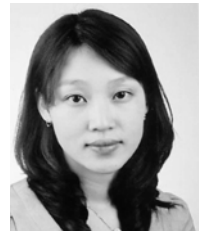
〔10〕 B. Lampson, V. Srinivasan, and G. Varghese, “IP Lookups Using Multiway and Multicolumn Search,” Proc. INFOCOM, pp. 1248-1256, Mar. 1998.

〔11〕 M. Buddhikot, S. Suri, and M. Waldvogel, “Space Decomposition Techniques for Fast Layer-4 Switching,” Protocols for High-Speed Networks, vol. 66, no. 6, pp. 277-283, Aug. 1999.

〔12〕 V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel, “Scalable Level 4 Switching and Fast Firewall Processing,” Proc. ACM SIGCOMM, pp. 203-214, Sep. 1998.

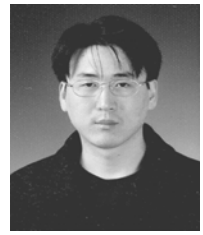
〔13〕 IDT, Network Search Engine(NSE) with QDR Interface, http://www.idt.com/.

정 해 진 (Haejin Jeong) 준회원



2001년 2월 한남대학교 컴퓨터공학과 학사 졸업
2004년 3월~현재 충남대학교 컴퓨터공학과 석사 과정
<관심분야> 컴퓨터 통신 및 보안, 네트워크프로세서

송 일 섭 (Ilseop Song) 준회원



2002년 2월 충남대학교 컴퓨터공학과 학사 졸업
2005년 2월 충남대학교 컴퓨터공학과 석사 졸업
2005년 3월~현재 충남대학교 컴퓨터공학과 박사 과정
<관심분야> 컴퓨터 통신 및 보안, 네트워크프로세서

이 유 경 (Yookyoung Lee) 정회원

1978년 2월 한국항공대학교 전자공학과 졸업
1980년 2월 연세대학교 대학원 전자공학과 석사
1984년 4월~현재 한국전자통신연구원 책임연구원, 팀장
<관심분야> 네트워기술, 이더넷, IP 네트워크, 네트워크프로세서

권 택 근 (Taeckgeun Kwon) 정회원

1988년 2월 서울대학교 컴퓨터공학과 학사 졸업
1990년 2월 서울대학교 컴퓨터공학과 석사 졸업
1996년 2월 서울대학교 컴퓨터공학과 박사 졸업
1998년 9월~현재 충남대학교 전기전자정보통신학부 부교수
<관심분야> 컴퓨터 통신 및 보안, 네트워크프로세서