情 報 保 護 學 會 誌
第 15 卷 第 6 號, 2005. 12

TCAM을 사용하는 보안 응용에서의 범위 검색 연구

김 영 호*, 김 기 영*, 장 종 수*

요 약

최근 초고속 네트워크의 보급에 따른 보안 장비의 성능 요구 사항은 점점 높아가고 있다. 특히 방화벽이나 침입탐지시스템에서 패킷과 보안 정책과의 일치 여부를 고속으로 알기 위해서 TCAM과 같은 하드웨어 기술이 점차 적용되고 있다. TCAM은 메모리 저장된 엔트리 중에서 입력키와 완전히 일치된 값을 찾거나 prefix 형태로 기술된 범위 내에 입력키가 어느 범위에 속하는 지 쉽게 찾을 수 있지만, prefix로 기술되지 않는 범위에 대해서는 하나의 엔트리로 표현하기 어렵다. 이렇듯 TCAM은 고속의 검색 기능을 제공하지만 다른 메모리 소자에 비해 가격이 비싸고 전력 소모가 크기 때문에 저장 공간의 낭비 없이 non-prefix 형태의 범위로 표현할 수 있는 방법이 요구된다. 본 논문에서는 범위변환 테이블을 이용하여 non-prefix 형태의 범위에 대해서도 prefix 형태의 범위와 동일하게 하나의 엔트리로 기술함으로써 저장 공간을 효율적으로 사용하고 다양한 보안 기능을 구현할 수 있는 가능성을 제시한다.

Ⅰ. 서 론

방화벽이나 침입탐지시스템은 외부로부터의 악의적인 공격으로부터 내부 네트워크나 호스트를 보호하기 위한 보안 장비들이다. 이 때 악의적인 공격 트래픽 또는 안전한 트래픽은 관리자에 의해 설정된 보안 정책에 따라 구분되며 패킷이 어떤 정책에 해당되는지 여부는 패킷의 정보에 따라 판단된다. 특히 패킷의 IP 헤더에 포함되어 있는 IP 주소와 프로토콜 번호 그리고 포트 번호는 이러한 방화벽이나 침입탐지시스템에서 패킷을 인식하는데 가장 많이 참조되는 정보들이다[1].

따라서 대부분의 보안 기능의 경우 기본적으로 패킷의 IP 헤더 정보를 이용하여 보안 정책인 규칙과 비교하는 연산 작업을 요구하고 있다. 뿐만 아니라, 최근 네트워크 장비들이 고속화됨에 따라 보안 장비의 고속화 및 정밀화가 요구되기 때문에 IP 헤더 정보를 규칙과 고속으로 정확하게 비교하는 검색 작업이 요구된다.

이러한 검색 작업을 소프트웨어적인 방법을 이용하여 구현한 기존 시스템에서는 입력된 규칙의 개수와 규칙들의 패턴에 따라 성능의 변화폭이 크게 나타나는 문제점을 지니게 되었다. 이러한 문제점을 해결하기 위해서 규칙 개수에 상관없이 일정하고 예측 가능한 성능을 보장하는 TCAM(Ternary Content Addressable Memory) 하드웨어 기술이 점차적으로 사용되고 있다. 이러한 TCAM 기술을 이용함으로써 완전(exact) 일치(match) 또는 prefix 형태의 범위 일치 검사가 고속으로 처리되고 규칙의 개수와 패턴에 상관없이 일정한 성능이 보장되게 되었다. 그러나 TCAM의 메모리 특성상 prefix 이외의 범위를 하나의 엔트리로 직접 기술할 수 없기 때문에 방화벽이나 침입탐지시스템 등의 보안 기능에서 non-prefix 범위 검색을 이용할 수 있도록 효율적인 범위 검색 알고리즘이 요구된다.[3]

Ⅱ. TCAM 기술의 특징 및 한계

일반적으로 TCAM은 입력 키 값을 메모리에 저장된 엔트리들과 내용을 기반으로 동시에 비교하여 일정한 시간 내에 그 중에서 일치하는 엔트리를 찾아내는데 이용되는 메모리의 한 종류이다. 이러한 TCAM을 이용한 검색은 메모리에 저장된 엔트리 중에서 키 값과 완전히 일치하는 값을 찾거나 don't care bit를 사용하여 키가 해당되는 prefix 범위를 갖는 엔트리를 찾는데 이용된다. 그러므로 완전히 일치하는 하나의 값 형태인 IP 주소나 prefix 형태의 범위로 기술

* 한국전자통신연구원 정보보호연구단 ({wtowto, kykim, jsjang}@etri.re.kr)

Source IP	Destination IP	Protocol	Source Port	Destination Port
10.1.0.0/16	24.1.0.0/16	TCP	any	1~14

00001010 00000001 xxxxxxxx xxxxxxxx	00011000 00000001 xxxxxxxx xxxxxxxx	00000110	xxxxxxxx xxxxxxxx	0000 0001
00001010 00000001 xxxxxxxx xxxxxxxx	00011000 00000001 xxxxxxxx xxxxxxxx	00000110	xxxxxxxx xxxxxxxx	0000 001x
00001010 00000001 xxxxxxxx xxxxxxxx	00011000 00000001 xxxxxxxx xxxxxxxx	00000110	xxxxxxxx xxxxxxxx	0000 01xx
00001010 00000001 xxxxxxxx xxxxxxxx	00011000 00000001 xxxxxxxx xxxxxxxx	00000110	xxxxxxxx xxxxxxxx	0000 10xx
00001010 00000001 xxxxxxxx xxxxxxxx	00011000 00000001 xxxxxxxx xxxxxxxx	00000110	xxxxxxxx xxxxxxxx	0000 110x
00001010 00000001 xxxxxxxx xxxxxxxx	00011000 00000001 xxxxxxxx xxxxxxxx	00000110	xxxxxxxx xxxxxxxx	0000 1110

[그림 1] TCAM에서 엔트리 확장을 이용한 범위 표현

되는 IP 주소는 하나의 TCAM 엔트리로 기술이 가능하기 때문에 라우팅 기술에서 고속으로 IP 주소 테이블 검색하여 다음 목적지를 찾는 데 적용되며 이는 TCAM이 이용되는 분야의 대표적인 예라 할 수 있다.[4][5]

그러나 don't care bit를 사용한 prefix 범위로 표현되지 않는 범위에 대해서는 하나의 TCAM 엔트리로 직접 표현하는 것이 불가능하기 때문에 prefix 이외의 범위 검색을 요구하는 응용에서는 하나의 범위를 여러 개의 TCAM 엔트리로 확장하여 표현하는 변환 기술을 사용할 수 있다.

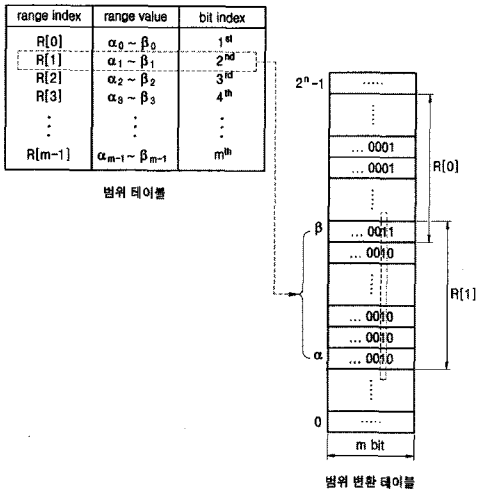
그림 1은 방화벽 시스템에서 흔히 볼 수 있는 규칙 정보의 예로서, 출발지 서브네트워크(10.1.0.0/16)에서 목적지 서브네트워크(24.1.0.0/16)로 전송되는 패킷 중에서 TCP 포트 1번부터 14번을 사용하는 패킷을 인식하는 규칙을 보여주고 있다. 이 규칙의 경우 IP 주소는 prefix 형태의 범위로 기술되어 있으며 TCP 목적지 포트 번호 범위(1~14)의 경우 prefix 형태의 범위가 아닌 non-prefix 형태의 범위로 기술되어 있다. 이 때 규칙을 TCAM의 엔트리로 표현하려면 non-prefix 범위로 인해서 하나의 엔트리로 직접 기술하지 못하고 그림 1에서 2진수 값으로 표현된 것처럼 최소한 6개의 엔트리로 확장해서 기술할 수 있다. 이론적으로 TCAM에서 n비트로 구성된 non-prefix 형태의 한 범위를 기술할 때, 최대 (2n-2)개까지의 엔트리로 확장될 수 있다. 그러므로 위의 예에서 TCP 목적지 포트 번호는 IP 헤더 내에서 16비트로 구성되기 때문에 TCP 목적지 포트 번호의 범위를 기술하는데 최대 30(2x16-2)개의 엔트리로 확장이 가능하다. 그러나 TCP 출발지 포트 번호도 위와 동일한 원리로 최대 30개의 엔트리로 확장될 수 있기 때문에 TCP 출발지 포트 번호와 TCP 목적지 포트 번호를 서로 다른 범위로 동시에 하나의 규칙으로 기술하는 경우, 하나의 규칙을 표현하기 위해서는 최대 900(30x30)개까지의 TCAM 엔트리를 필요로 할 수 있게 된다. 이 경우 non-prefix 형태의 범위를 기술하기 위한 단순 확장 방법은 메모리의 저장 공간 이용률을 0.1% 수준으로 유지하기 때문에 실제 상용 제품에 적용하기 부적합하다.

TCAM은 다른 메모리 소자에 비해 단위당 가격이 비싸고 전력 소모량이 많기 때문에 보다 효율적인 관리를 위해 현재 다양한 연구가 진행되고 있다. 예를 들어, 전체 TCAM에 전원을 모두 공급하지 않고 TCAM을 여러 세그먼트로 분리하고 실제로 엔트리가 들어 있는 활성화된 세그먼트에만 전원을 공급시켜 불필요한 전력 소모를 막고 보다 효율적으로 관리할 수 있는 방법이 제안되고 있다. 그러므로 효율적인 전력 사용 및 제품의 생산 비용을 줄이기 위해서는 먼저 효율적인 저장 공간 이용이 선행되어야 한다.

그러므로 non-prefix 형태인 하나의 범위를 표현하기 위해서 여러 개의 엔트리로 확장시키는 방법은 TCAM의 저장 공간 측면에서 효율성이 떨어지게 하고 이는 불필요한 전력 낭비를 초래하게 하며 또한 최종 제품의 가격상승의 원인이 될 수 있다. 이에 본 논문에서는 모든 범위를 하나의 엔트리로 표현하기 위해서 간접적인 범위 변환 방법을 제안하고 있다.

Ⅲ. 범위 변환

TCAM에서 prefix 형태의 범위뿐만 아니라 non-prefix 형태의 범위에 대해서도 하나의 엔트리로 표현하기 위해서 non-prefix 형태의 범위를 특정 패턴 또는 값으로 변환하고 변환된 값을 TCAM의 엔트리로 표현한다. 이를 위해 본 논문에서 제안하는 범위 변환 방법은 크게 범위 테이블과 범위변환 테이블로 구성된다.



〔그림 2〕 범위 테이블 및 범위변환 테이블

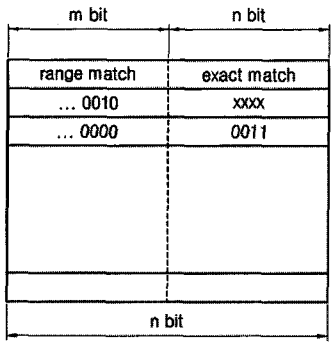
3.1. 범위 변환 테이블

범위 테이블은 중첩 가능한 m 개의 non-prefix 형태의 범위를 m 비트로 표현함에 있어서, 각각의 범위에 대하여 m 비트 중 '1'로 설정되는 특정 비트의 위치 정보를 포함한다. 예를 들어, 범위 테이블에서 non-prefix 형태의 범위를 표현하는 m 비트의 첫 번째 비트가 '1'로 설정된 경우 α0~β0 범위(R[0])를 나타내고, m 비트의 두 번째 비트가 '1'로 설정된 경우 α1~β1 범위(R[1])를 나타낸다. 여기서 non-prefix 형태의 서로 다른 두 범위는 중첩될 수 있다. 만약 새로 추가되는 규칙에 non-prefix 형태의 범위 검색을 이용하는 부분이 포함된 경우, 범위 테이블에 새로운 범위 항목을 할당하고 추가된 범위 항목에 해당하는 비트 인덱스를 가지고 이후 범위변환 테이블에 적용한다.

범위 변환 테이블은 m 개의 non-prefix 형태의 범위를 동시에 지정할 수 있는 2^n 개의 m 비트 엔트리로 구성된다. 예를 들어, 새로 추가되는 범위 검색을 위한 엔트리의 범위가 $\alpha_1 \sim \beta_1$이고, 범위 테이블 내에서 두 번째 엔트리로 추가되면, 그림 2와 같이 해당 범위($\alpha_1 \sim \beta_1$)에 속하는 범위 변환 테이블의 모든 엔트리들의 두 번째 비트를 1로 세팅한다. 반대로, 특정 non-prefix 형태의 범위가 삭제되는 경우에는 범위 테이블에서 해당 범위에 속하는 모든 범위변환 테이블 엔트리들의 해당 범위 인덱스 비트를 0으로 변경한다.

3.2. TCAM 엔트리

본 논문에서 제안하는 범위 검색에서는 non-prefix 형태의 범위 검색뿐만 아니라 동시에 완전 일치 검색 기능도 동시에 제공한다. 이를 위해 TCAM 엔트리는 완전 일치 검색을 위한 부분과 non-prefix 형태의 범위 검색을 위한 부분을 동시에 포함한다. 그림 3은 이러한 내용을 반영한 TCAM의 엔트리 구조를 보여주고 있다.



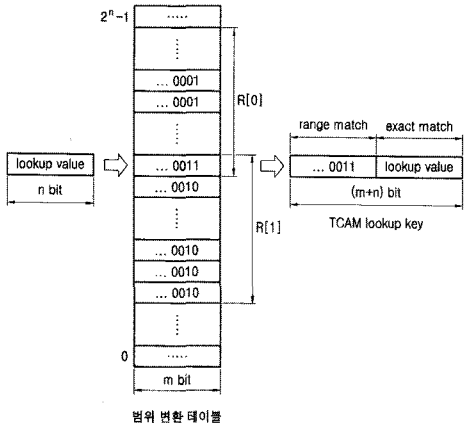
〔그림 3〕 TCAM 엔트리

TCAM 엔트리 구조에서 완전 일치 검색을 위한 엔트리는 상위 m 비트의 범위 기술 부분이 0으로 이루어지고 나머지 n 비트로 하나의 완전 매칭 값을 정의한다. 반면 non-prefix 형태의 범위 검색을 위한 엔트리는 범위 테이블에 의해 정의된 비트 인덱스 값으로 상위 m 비트의 값을 구성하고 나머지 하위 n 비트는 don't care bit(x)로 기술한다.

이렇게 새로운 규칙이 추가되는 경우 (m+n) 비트로 이루어진 새로운 TCAM 엔트리가 TCAM 테이블에 추가된다. 이후에 TCAM에 대한 검색을 수행하면 완전 일치 검색뿐만 아니라 non-prefix 형태의 범위에 대한 검색도 동시에 수행할 수 있게 된다.

이미 저장된 규칙에 대한 검색을 수행할 때에는 n 비트의 검색 값을 인덱스로 하여 앞에서의 범위변환 테이블로부터 추출한 엔트리의 값으로 최종 검색 키의 상위 m 비트의 값으로 구성하고, 초기 검색 값을 최종 검색 키의 하위 n 비트의 값으로 구성한다. 이렇게 생성된 최종 검색 키를 이용하여 TCAM 테이블 내에서 해당하는 TCAM 엔트리를 찾는다.

그림 4는 TCAM 엔트리에서 검색 값(lookup value)이 입력되면 검색 값에 해당하는 범위변환 테이블의 엔트리를 추출하고, 그 추출한 범위변환 테이블의 엔트리 값을 최종 검색 키의 상위 m 비트의 값으로 구성하고, 하위 n 비트의 값은 입력 값을 그대로 적용하여 (m+n) 비트의 최종 TCAM 검색 키(lookup



〔그림 4〕 검색키 변환과정

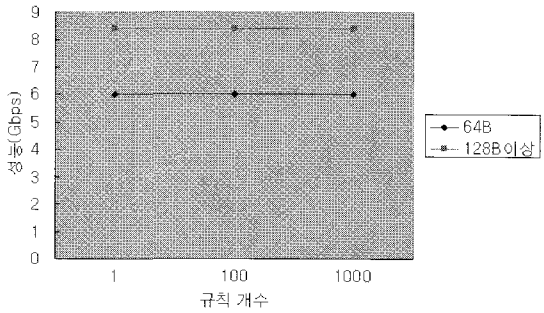
key)를 생성하는 과정을 보여준다.

범위 변환 테이블을 이용한 non-prefix 형태의 범위 검색 방법은 범위변환 테이블의 값을 읽는 과정을 부가적으로 수행하기 때문에 기존 확장된 엔트리를 직접 TCAM에 기술한 방법에 비해서 전체 검색 시간이 더 소요된다. 그러므로 본 논문에서 제안한 범위 검색 방법은 TCAM 저장 공간의 효율성을 높이기 위해서 검색 시간을 지연시키는 단점이 있다. 그러나 실제 환경에서의 구현 및 테스트를 통하여 저장 공간의 효율성을 높이기 위해 희생된 검색 시간이 충분히 무시될 수 있음을 보여주고 있다..

Ⅳ. 구현 및 분석

지금까지 살펴본 범위 검색 방법을 실제 환경에 적용하여 검증하기 위해서 Intel IXP2800 네트워크프로세서 플랫폼 상에 구현하였다. 이 때 TCAM은 IDT사의 IDT75K62134 9Mb 모델을 사용하였다.

Intel IXP2800은 OC-192/10 Gbps의 성능을 갖는 제2세대 네트워크 프로세서로, XScale 코어와 16개의 마이크로엔진으로 구성된다. 특히 마이크로엔진 내에는 고속으로 패킷을 처리할 수 있는 하드웨어 쓰레드를 제공하고 있으며 마이크로엔진 간에는 병렬처리 및 파이프라인처리를 통해 고속의 패킷 처리가 가능하다. 또한 다중쓰레드 기법은 대부분의 네트워크 트래픽 처리가 CPU 보다는 I/O 처리에 집중되어 있기 때문에 I/O로 인한 지연 시간을 다른 쓰레드의 실행 시간으로 감추는 기법(I/O Latency Hiding)을 통하여 성능을 최적화한다.[2]



〔그림 5〕 패킷필터링 성능(loss tolerance<0.00001%)

구현은 방화벽 기술의 기반이 되는 5-tuple 패킷 필터링 기능을 논문에서 제안하고 있는 범위 검색 방법을 적용하여 마이크로코드로 구현하였으며 성능 분석기인 IXIA 장비를 이용하여 양방향 트래픽으로 성능을 측정하였다. 이 때 패킷필터링 규칙을 1개부터 1000개까지 추가하면서 성능을 측정하였으며 non-prefix 형태의 범위를 기술하는 규칙이 반드시 포함되도록 하였다.

그림 5는 측정된 5-tuple 패킷필터링의 성능으로 64B 패킷의 경우 6Gbps, 128B 이상의 패킷에서는 8.4Gbps로 패킷필터링 추가 전의 성능과 동일한 결과를 보였다. 따라서 범위 변환 과정으로 인한 부가적인 메모리 접근이 전체 성능에 큰 영향을 주지 않음을 보여준다. 그러므로 본 논문에서 제안한 non-prefix 형태의 범위 검색 방법은 TCAM의 성능 효과를 크게 저하시키지 않으면서 동시에 저장 공간의 효율성을 높여주고 있다.

Ⅴ. 결 론

네트워크의 고속화에 따라 보안 장비에서도 TCAM과 같은 하드웨어 기술의 점차 도입되고 있다. 그러나 TCAM은 non-prefix 형태의 범위를 직접 기술할 수 없기 때문에 저장 공간 및 전력 효율을 높이기 위한 새로운 범위 검색 기술이 요구된다. 본 논문에서 제안한 non-prefix 형태의 범위 검색 방법은 하나의 범위를 기술하기 위하여 하나의 TCAM 엔트리만을 사용하기 때문에 부가적인 TCAM 엔트리를 요구하지 않는다. 이는 저장 공간의 효율성을 높여 불필요한 비용을 없애고 소모 전력을 줄이는 효과가 있다. 또한 실제 플랫폼 상에서의 구현과 실험을 통해 범위변환 테이블로 인한 성능 저하가 전체 시스템 성능에 크게 영향을 주지 않음을 보여주고 있다. 이러한 non-prefix

형태의 범위 검색 방법은 보안 응용뿐만 아니라 고속의 범위 검색을 요구하는 다른 응용 분야에서도 적용이 가능하다.

참 고 문 헌

[1] Robert Zalenski, "Firewall Technologies", March, 2002.

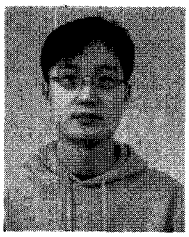
[2] Eric J. Johnson, Aaron R. Kunze, "IXP2400/2800 Programming", Intel Press, April 2003.

[3] Huan Liu, "Efficient Matching of Range Classifier into Ternary-CAM," *Proceed ings of the 10th Symposium on High Performance Interconnects HOT Interconnects*, 2002.

[4] Zhijun Wang, Hao Che, Mohan Kumar, "CoPTUA: Consistent Policy Table Update Algorithm for TCAM without Locking", IEEE Transactions on Computers, Vol. 53, December 2004.

[5] D. Shah, P. Gupta, "Fast incremental updates on Ternary-CAMs for routing lookups and packet classification", Proc. IEEE Hot Interconnects VIII, August 2000.

〈著 者 紹 介〉

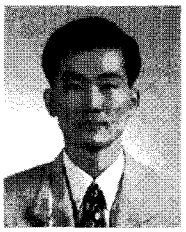


김 영 호 (Youngho Kim)

1999년 2월 : 고려대학교 컴퓨터학과 졸업
2001년 2월 : 고려대학교 컴퓨터학과 석사
2002년 11월~현재 : 한국전자통신연구원 연구원
〈관심분야〉 운영체제, 네트워크보안, IPv6 보안

김 기 영 (Kiyoung Kim)

1988년 2월 : 전남대학교 전산통계학과 졸업
1993년 2월 : 전남대학교 전산통계학과 석사
2002년 2월 : 충북대학교 전자계산학과 박사
1988년 2월~현재 : 한국전자통신연구원 보안운영체연구팀 팀장/책임연구원
〈관심분야〉 네트워크보안, 고성능 네트워크 침해탐지 및 대응기술, IPv6 보안기술



장 종 수 (Jongsoo Jang)

1984년 2월 : 경북대학교 전자공학과 졸업
1986년 2월 : 경북대학교 전자공학과 석사
2000년 2월 : 충북대학교 컴퓨터공학과 박사
1989년 7월~현재 : 한국전자통신연구원 네트워크보안그룹장/책임연구원
〈관심분야〉 네트워크보안, 고성능 네트워크 침해탐지 및 대응기술, 정보보호