네트워크 프로세서와 TCAM을 이용한 패킷 필터링 시스템 설계 및 구현[1]

강석민° 정해진 권택근
충남대학교 컴퓨터공학과
esemkang°@gmail.com, {hjjeung, tgkwon}@cnu.ac.kr

The Design and Implementation of the Packet Filtering System using Network Processor and TCAM

Seok-Min Kang° Hae-Jin Jung, Teack-Geun Kwon
Department of Computer Engineering, Chungnam National University

요 약

네트워크 관련 기술 발전의 순기능으로 네트워크 고속화, 사용자 증가, 그리고 다양한 응용의 발생 및 적용 등을 생각할 수 있다. 그러나 그 역기능으로 네트워크를 통한 바이러스, 인터넷 웜의 전파, 불법적인 시스템 접근 등 다양한 방식의 크래킹 기법들이 늘어나고 있음을 알 수 있다. 본 논문에서는 이런 네트워크를 이용한 공격의 대응 방식 중 하나인 패킷 필터링 기능을 네트워크 프로세서 기반의 10Gb 고속 라우터에 설계 및 구현함으로써 라우터 단에서의 패킷 필터링 기능 제공에 대해 기술하고 있다.

1. 서 론

네트워크 환경은 점차 고속화되고, 수용할 수 있는 네트워크 사용자의 수도 대폭 증가하고 있다. 또한 급증하는 네트워크 사용자들의 요구에 부응하여 관련 응용들도 다양한 분야에서 발굴되어 적용되고 있다. 하지만 그 이면에는 크래킹 기술의 발달과, 급속히 전파되어 네트워크에 연결된 불특정 다수의 단말들에 피해를 입히는 인터넷 웜을 비롯해 다양한 형태의 공격들도 함께 증가하고 있다. 이런 네트워크를 이용한 다양한 공격에 대응하는 방식 중 하나로 모든 패킷들을 검사하여 안전/불안전 여부에 따라 패킷을 수용하거나 거부하는 방화벽 시스템이 있다.

본 논문에서는 네트워크 프로세서를 이용한 10Gb 고속 라우터에 패킷 필터링 기능을 설계하고 마이크로 프로그래밍을 통해 구현함으로써 라우터 단에서 고속으로 패킷을 처리하면서 방화벽 기능을 수행하는 방법에 대해 기술하고 있다.

2. IXP2850 기반의 패킷 필터링

본 논문에서 구현한, 인텔 네트워크 프로세서 IXP2850을 이용한 패킷 필터링 시스템은 패킷 필터링을 위한 규칙을 관리하는 제어부 (Control Plane)와 고속으로 패킷을 처리하면서 패킷을 검사하고 허용/거부를 결정하는 전송부 (Forwarding Plane)로 구성된다. 자세한 구조는 다음의 [그림 1]과 같다.

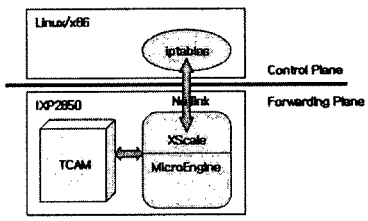

그림 1. 패킷 필터링 시스템 구조

2.1 제어부

제어부는 x86 기반의 범용 프로세서 상에 리눅스를 구동하는 전형적인 개인용 컴퓨터의 형태이다. 제어부는 사용자로부터 패킷 필터링을 위한 규칙을 입력 받고, 그 규칙을 넷링크(Netlink) 형태의 IPC (Inter-Process Communication) 메시지로 변환하여 전송부로 내려주는 역할을 담당한다 [1]. 본 논문에서는 리눅스의 iptables v1.2.9 패키지를 기반으로, 사용자가 규칙을 추가하는 경우 그 규칙을 넷링크 IPC 메시지 형태로 변환하고 전송부

[1] 이 연구는 BK21충남대학교 정보통신인력양성사업단의 지원을 받았음.

로 보내는 모듈을 추가하였다. 리눅스 iptables 는 다양한 필터링 규칙을 제공해주는데, 각 옵션의 기능 설명은 2.2.3 절에서 설명한다 [2].

2.2 전송부

전송부에서는 인텔 네트워크 프로세서 IXP2850 두 개를 사용하여 10Gbps 의 처리가 가능한 인텔 IXDP2800 개발 플랫폼을 이용하였다. 두 개의 네트워크 프로세서는 각각 ingress와 egress를 담당하게 된다. 이 절에서는 그 중 패킷 필터링 기능을 담당하는 ingress를 두 부분으로 나누어 기술한다 [3].

2.2.1 XScale

XScale 코어는 ARM V5TE 구조와 호환되는 프로그램 가능한 프로세서이다. XScale 코어에 리눅스를 포팅하여 제어부가 전송한 넷링크 IPC 메시지를 분석하고 그 규칙을 유지/관리하는 기능을 담당한다. 규칙은 TCAM (Ternary-CAM)에 저장된다.

2.2.2 마이크로 엔진

마이크로 엔진은 패킷을 수신하고, 수신한 패킷이 해당하는 규칙이 있는지 검사하고, 허용된 패킷을 라우팅하고 송신하는 역할을 담당한다. 각 마이크로 엔진의 기능은 마이크로 코드를 이용하여 구현되었다. 인텔 네트워크 프로세서 하나에는 16개의 마이크로 엔진이 내장되어 있다. 각 마이크로 엔진에 할당된 소프트웨어 모듈은 다음의 [그림 2]와 같다 [4,5].

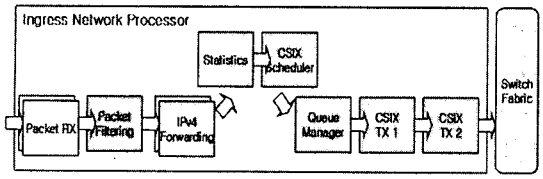

그림 2. Ingress NP 마이크로 엔진 S/W 구조

[그림 2]의 패킷 필터링 모듈이 패킷을 처리하는 과정은 다음의 [그림 3]과 같다.

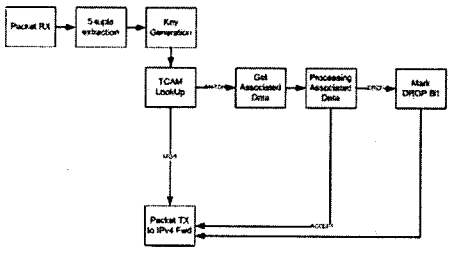

그림 3. 패킷 필터링 모듈 흐름도

패킷 수신 모듈이 패킷을 수신하면, DRAM에 저장한 후 제어를 패킷 필터링 모듈로 넘긴다. 패킷 필터링 모듈은 수신한 패킷의 5-튜플 정보 (송신 주소, 송신 포트, 수신 주소, 수신 포트, 프로토콜)를 추출하고 이 값을 키(Key)로 하여 필터링 규칙을 저장한 TCAM을 검색하게 된다. 패킷에 해당하는 규칙이 발견되면, 그 규칙에 연계된 정보 (associated data)를 SRAM으로부터 읽어오고 패킷의 허용/거부 여부를 판별한다. 판별이 되면 패킷의 메타정보에 허용/거부 여부를 표시하고, IPv4 포워딩 모듈에게 패킷의 메타정보를 넘겨주게 된다. IPv4 포워딩 모듈은 허용 표시가 된 패킷만 포워딩을 하고, 거부 표시가 된 패킷은 버리게 된다.

2.2.3 패킷 필터링 기능

본 논문의 패킷 필터링 시스템은 2.1절에서 언급한대로 iptables를 참고하여 마이크로 코드로 구현하였다. 각 기능별 설명은 다음과 같다 [6].

- 플로우 별 패킷 필터링

IP 기반의 패킷들을 5-튜플 정보를 기준으로 플로우 단위로 판별한 후 필터링을 한다. (예, tcp 192.168.10.0/24 * 192.168.10.0/24 0:1024 DROP).

- 패킷 정보에 따른 필터링

TCP 플래그는 TCP 세션을 제어하기 위한 제어필드이다. 이 제어필드의 정보를 이용한 필터링 기능은 최근 빈번히 나타나는 공격 기법인 SYN 플러딩 등의 DDoS (Distributed Denial of Service) 공격에 대처하기 위한 효율적인 수단이다.

- TCP 플로우 상태 정보를 이용한 필터링

TCP 트래픽은 IP 망에서 큰 비중을 차지한다. 즉, TCP 플로우에 대한 제어의 중요성이 높다고 할 수 있다. 이 기능은 TCP 플로우의 상태 - 연결이 시작되는 상태 (NEW), 연결된 상태 (ESTABLISHED) - 에 따른 필터링을 지원한다.

- 전송률에 따른 필터링

본 논문의 패킷 필터링 기능은 단위 시간당 허용할 패킷의 수를 제한하는 기능을 제공한다. 패킷 도착 순간의 타임 스탬프 값을 이용하여 규칙에 설정한 수의 패킷만을 허용할 수 있다.

2.2.4 TCAM (Ternary-CAM)

TCAM은, 기존의 CAM이 0, 1 을 지원하는데 더해서 " don' t care" 비트를 지원한다. 인텔 네트워크 프로세서 IXP2850에서는 SRAM 4번째 채널에 SRAM과 TCAM이 동시에 연결되어 있으므로 4번째 채널에 접근함으로써 TCAM 접근이 가능하다 [7-8].

본 논문의 패킷 필터링 기능은 필터링을 위한 규칙 테이블 접근 시 TCAM을 이용함으로써 한번의 TCAM 검색으로

패킷과 규칙의 매치 여부를 결정할 수 있다. TCAM의 접근 시간은 보통 5ns 이하로, 매우 빠른 시간 내에 필터링 규칙 검색을 끝내는 장점이 있다. 추가로, TCAM의 검색 결과에 대한 관련 정보 (associated data)를 얻어오기 위해, TCAM 검색 결과를 인덱스로 SRAM을 한번 접근하여 정보를 추출한다 [9-10]. 다음의 [그림 4]는 TCAM에 저장되는 5-튜플 정보를 이용한 키와 그에 대한 마스크 구조를 보여준다.

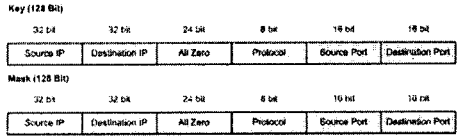

그림 4. TCAM의 키와 마스크 구조

3. 실험 환경 및 결과

본 논문에서는 인텔 IXDP2800 개발 플랫폼과, 패킷 생성을 위해 스마트비트를 사용하였다. 다음의 [그림 5]는 개발 환경을 보여준다.

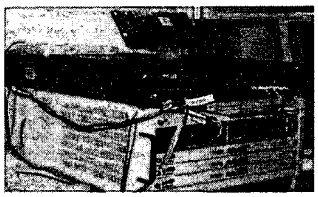

그림 5. 인텔 IXDP2850과 스마트비트

본 논문의 패킷 필터링 시스템은 ingress 네트워크 프로세서의 마이크로 엔진 한 개를 할당하고, 8개의 쓰레드 중 한 개의 쓰레드를 사용하여 구현하였다. 스마트비트를 이용하여 1Gbps의 트래픽을 생성하여 성능을 측정한 결과는 다음의 [그림 8]과 같다.

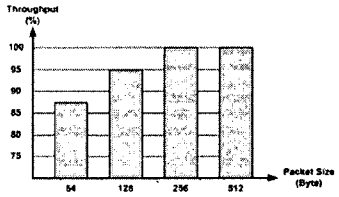

그림 6. 패킷 필터링 성능 측정 결과

64바이트 패킷을 생성했을 경우 87.5%, 128바이트 인 경우는 95%의 처리율을 보인다. 256바이트와 512바이트의 경우 100%의 처리율을 보임을 알 수 있다. 64, 128바이트 패킷 실험 시 성능저하가 나타나는데, 이는 하나의 마이크로 엔진에 하나의 쓰레드만을 이용했기 때문이다. 추후 성능 향상을 위한 논의는 4장에서 논의하고자 한다.

4. 결론 및 향후 연구 과제

본 논문에서는 인텔 네트워크 프로세서 IXP2850을 사용하여 고속 라우터에 패킷 필터링 기능을 구현한 내용을 기술하였다. 패킷 필터링 기능은 플로우 단위, 패킷 정보 단위, TCP 플로우 상태 단위, 그리고 전송률에 따라 규칙을 설정할 수 있으며 TCAM을 사용하여 규칙 검색을 위한 메모리 접근 시간을 단축시켰다.

3장의 실험 결과에서 하나의 마이크로 엔진과 하나의 쓰레드로는 10Gbps의 성능을 낼 수 없음을 알았다. 추후 마이크로 엔진 두 개를 이용하고, 총 16개의 쓰레드를 파이프-라인 형태로 구현하면 10Gbps의 성능을 보장할 수 있다.

참고문헌

[1] J. Salim, H. Khosravi, A. Kleen, A. Kuznetsov, Linux Netlink as an IP Services Protocol, RFC3549, 2003.

[2] Rusty Russell, " Linux 2.4 Packet Filtering HOWTO ", http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html, 2004.

[3] Intel, Intel® IXP2800/IXP2850 Network Processor : Hardware Reference Manual, 2002.

[4] Intel, " Intel® IXP2850 Network Processor", http://www.intel.com/design/network/products/npfamily/ixp2850.htm, 2004.

[5] B. Carlson, " Intel® Internet Exchange Architecture and Applications: A Practical Guide to IXP2XXX Network Processors" , Intel Press, 2003.

[6] E. J. Johnson and A. R. Kunze, " IXP2400/2800 Programming: The Complete Microengine Coding Guide" , Intel Press, 2003.

[7] Netlogic microsystems. Ternary Synchronous Content Addressable Memory (IPCAM). http://www.netlogicmicro.com/pdf/NL82721.pdf, 2003.

[8] IDT, IXP Lookup Management Library (LML) Programmer' s Guide, 2002

[9] E. Spitznagel, D. Taylor, and J. Turner, " Packet Classification Using Extended TCAMs" , ICNP, November 2003

[10] H. Liu, " Reducing Routing Table Size Using Ternary-CAM" , Hot Interconnects, August 2001