IPv6 네크워크 환경에서의 비정상 트래픽 제어 프레임워크 설계

김가을*, 강성구*, 김재광*, 고광선*, 강용혁**, 엄영익*
*성균관대학교 컴퓨터공학과
**극동대학교 경영학부
e-mail:{zfall, lived, linux, rilla91, yieom}@ece.skku.ac.kr,
yhkang@mail.kdu.ac.kr

A Design of an Abnormal Traffic Control Framework in IPv6 Network

Kaeul Kim*, Seong-Goo Kang*, Jaekwang Kim*,
Kwangsun Ko*, Young-hyeok Kang**, and Young Ik Eom*
*School of Info. and Comm. Eng., Sungkyunkwan University
**Dept of Business Administration, Far East University

요 약

IPv4 프로토콜의 주소 고갈 문제를 해결하기 위하여 IPv6 프로토콜이 제안되었고 한국전산원의 발표에 의하면 2010년 이후에는 IPv6 프로토콜이 광범위하게 사용될 것이라고 한다. 이러한 IPv6 프로토콜은 IPv4 프로토콜의 단점들을 해결하기 위하여 ND(Neighbor Discovery) 메커니즘, 주소자동설정, IPsec 등의 기술을 지원하며, 특히 IPv6 프로토콜은 보안 문제를 해결하기 위해서 인증, 데이터 무결성 보호를 위한 IPsec 기술을 사용한다. 이러한 IPsec 기술은 패킷 정보를 보호하기 위한 목적으로 사용되기 때문에 불특정 다수의 사용자를 대상으로 하는 네트워크에 행해지는 분산 서비스 거부 공격과 같은 비정상 대용량 트래픽에 대한 탐지 및 차단에 어려움이 있다. 현재 IPv6 프로토콜을 지원하는 네트워크 공격 대응 기술로 IPv6 네트워크용 방화벽/침입탐지 시스템이 개발되어 제품으로 판매되고 있지만, 대용량의 비정상 트래픽 대응 기술을 탐지하고 차단하기에는 한계가 있다. 본 논문에서는 IPv6 네트워크 환경에서 이러한 대용량의 비정상 트래픽을 제어할 수 있는 프레임워크를 제시한다.

1. 서 론

IPv4 프로토콜의 주소 고갈 문제를 해결하기 위하여 1993년 IPv6 프로토콜이 제안되었으며 한국 전산원 보고에 따르면 2010년 이후에는 IPv6 프로토콜이 광범위하게 사용될 것이라고 한다. IPv6 프로토콜은 IPv4 프로토콜의 여러 가지 문제점들을 해결하기 위해 주소자동설정, ND(Neighbor Discovery) 메커니즘, IPsec등과 같은 다수의 새로운 메커니즘을 도입했다[1][2].

특히 IPv6 프로토콜은 보안 문제를 해결하기 위하여 IPsec 메커니즘을 사용할 수 있으나, IPsec 메커니즘은 상호간의 인증과 신뢰성 있는 메시지 교환을 위한 메커니즘이기 때문에 대용량의 비정상 트래픽을 탐지하고 차단하는 데에는 한계가 있다. 또한, 최근 개발되어 제품으로 판매되고 있는 IPv6 프로토콜을 지원하는 방화벽/침입탐지 시스템들은 이러한 대용량의 비정상 트래픽을 탐지하고 차단하기에는 한계가 있다.

본 논문에서는 IPv6 네트워크 환경의 Linux 시스템에서 실시간 패킷 분석 기능과 유연한 제어 정책을 제공하기 위해여 netfilter 프레임워크와 CBQ 메커니즘을 기반으로 커널레벨에서 동작하는 비정상 트래픽 제어 프레임워크를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 비정상 트래픽 제어 프레임워크를 이해하는데 필요한 배경 지식을 소개하고, 3장에서 전체 구성과 구성 모듈별 상세 내용을 설명한다. 4장에서는 제안 프레임워크를 테스트하며, 5장에서는 IPv6 네트워크용 방화벽과의 기능비교를 실시하며, 마지막 6장에서는 결론 및 향후 연구 내용에 대해 기술한다.

2. 배경 지식

본 장에서는 IPv6 네트워크 환경에서 비정상 트래픽 제어 프레임워크 설계에 필요한 배경 지식들을 살펴보도록 한다.

2.1 IPv6 네트워크의 특징

IPv6 프로토콜은 IPv4 프로토콜의 주소 고갈 문제 해결을 포함해, 최근 그 중요도가 부각되고 있는 라우팅의 효율성, 보안 기능, 이동성 지원, QoS 보장 등을 목표로 탄생하게 되었다. IPv6 프로토콜은 이러한 장점 때문에 기존 IPv4 프로토콜을 대체할 것으로 보인다. IPv6 프로토콜은 IPsec을 이용하여 인증, 보안 암호화, 데이터 무결성 보호 등의 기능을 제공하지만, 대용량의 트래픽을 이용하는 공격에 대한 탐지 및 대응을 하기는 어렵다[3].

2.2 IPv6 기반 방화벽 및 침입탐지 시스템

침입이란 시스템 자원에 대한 무결성, 기밀성 또는 가용성을 침해하는 행위를 의미하며 방화벽/침입탐지 시스템이란 이러한 비인가된 사용자로부터의 침입을 탐지하여

시스템 자원을 보호하는 역할을 한다. 현재 IPv6 환경을 위한 방화벽 및 침입탐지 시스템이 개발되어 판매되고 있지만, 이러한 방화벽은 IPsec 기술을 사용한 패킷을 열어볼 수 없기 때문에 IPsec 기술을 악용한 대용량의 비정상 트래픽에 대한 대응을 할 수 없다.

2.3 Netfilter 프레임워크와 CBQ 메커니즘

본 논문에서는 패킷을 실시간으로 처리하기 위해서 리눅스 커널에서 지원하는 netfilter 프레임워크를 사용한다. Netfilter 프레임워크는 5개의 훅(hook) 포인트를 가지게 되는데, 패킷이 IP 계층으로 들어오면 이 훅 포인트를 지나가게 되고 각 포인트에 적절한 처리를 등록해서 들어온 패킷을 무시하거나 통과시키는 패킷 필터링을 실시간으로 처리한다. IPv6 네트워크 환경에서는 netfilter 프레임워크를 수정한 netfilter6 프레임워크가 사용된다[4][5][6].

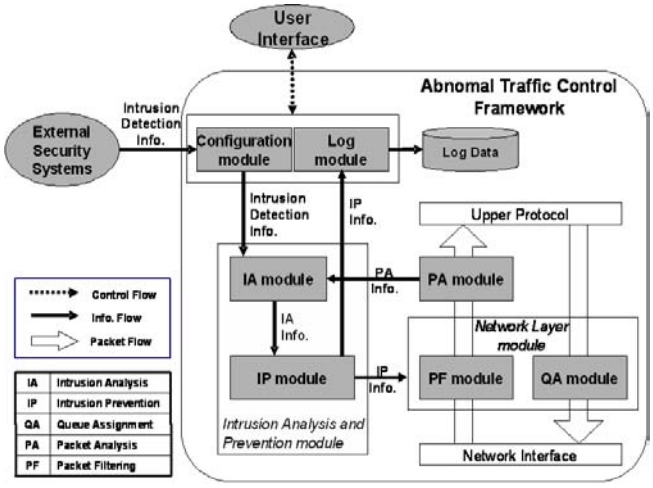
오탐지율(False Positive)은 침입탐지 시스템의 문제점 중 하나로 지적되어 왔다. 이러한 오탐지율을 줄이기 위해 본 논문에서는 CBQ 메커니즘을 사용하여 하나의 출력 큐 대신에 여러 개의 출력 큐를 클래스 별로 두어서 우선순위를 정하고 각 큐 별로 서비스 되는 트래픽의 양을 조절한다.

3. 프레임 워크 설계

본 장에서는 IPv6 네트워크 환경에서 리눅스 시스템을 기반으로 커널 모드에서 동작하는 비정상 트래픽 제어 프레임워크 설계를 제안한다.

3.1 전체 구성도

리눅스 시스템 기반의 비정상 트래픽 제어 프레임워크의 전체적인 모듈 구성은 그림 1과 같다.



(그림 1) 제안 시스템 전체 모듈 구성도

프레임워크는 네트워크 인터페이스로부터 전달된 패킷을 PF 모듈에서 필터링 룰 셋(Rule Set)과 비교하여 패킷을 차단할지 상위 프로토콜로 전달할지 결정하고, 차단되지 않은 패킷의 정보는 PA 모듈을 통하여 IA 모듈로 전달한다. IA 모듈에서는 전달된 트래픽의 정보는 비정상적인 패킷인가를 검사한 후, 비정상적이라고 판단되면 IP 모듈에 트래픽 정보를 전달한다. IP 모듈은 CBQ 메커니즘을 이용하여 비정상 트래픽의 대역폭을 줄이거나 비정상 트래픽 정보를 필터링 룰 셋에 추가한다. 추후 제한된 대역폭을 가진 호스트가 재차 침입으로 의심되는 패킷을 전송할 경우 IP 모듈에서 필터링 룰 셋에 호스트 정보를 추가하여 PF 모듈에서 차단하게 한다.

3.2 IPv6 환경을 위한 고려사항

IPv6 네트워크를 대상으로 하는 비정상 트래픽 제어 프레임워크를 설계하는데 있어서 3가지 고려사항이 있다.

```
struct iphdr {
#if defined(__LITTLE_ENDIAN_BITFIELD)
        __u8    ihl:4,
                version:4;
#elif defined (__BIG_ENDIAN_BITFIELD)
        __u8    version:4,
                ihl:4;
#else
#error  "Please fix <asm/byteorder.h>"
#endif
        __u8    tos
        __u16   tot_len
        __u16   id;
        __u16   frag_off
        __u8    ttl
        __u8    protocol;
        __u16   check;
        __u32   saddr
        __u32   daddr
        /*The options start here. */
};
```

(그림 2) IPv4 헤더 구조 (include/linux/ip.h)

```
struct ipv6hdr {
#if defined(__LITTLE_ENDIAN_BITFIELD)
        __u8                    priority:4,
                                version:4;
#elif defined(__BIG_ENDIAN_BITFIELD)
        __u8                    version:4,
                                priority:4;
#else
#error  "Please fix <asm/byteorder.h>"
#endif
        __u8                    flow_lbl[3];
        __u16                   payload_len
        __u8                    nexthdr
        __u8                    hop_limit

        struct  in6_addr   saddr
        struct  in6_addr   daddr
};
/*      IPv6 address structure          */
struct in6_addr
{
   union
   {
      __u8    u6_addr8[16];
      __u16   u6_addr16[8];
      __u32   u6_addr32[4];
   } in6_u;
#define s6_addr    in6_u.u6_addr8
#define s6_addr16  in6_u.u6_addr16
#define s6_addr32  in6_u.u6_addr32
};
```

(그림 3) IPv6 헤더 구조 (include/linux/ip6.h)

첫 번째 고려 사항은 IPv4 프로토콜의 주소공간을 처리하기 위한 구조체 대신 IPv6 프로토콜의 주소공간을 처리하기 위한 구조체를 사용하는 것이다. IPv4 프로토콜은 32bit의 주소공간을 가지고 있고 IPv6 프로토콜은 128bit의 주소공간을 가지고 있는 것을 그림 2과 그림 3의 IPv4와 IPv6 헤더 구조를 통해 알 수 있다.

두 번째 고려사항은 IPv6 네트워크 환경을 위하여 IPv6 프로토콜을 지원하는 netfilter 프레임워크를 사용해야 한다는 것이다. 즉, 기존의 IPv4 프로토콜을 지원하는 netfilter 프레임워크 대신 IPv6 프로토콜을 지원하는

netfilter6 프레임워크를 사용한다. IPv4 프로토콜과 IPv6 프로토콜의 헤더내용과 주소 공간 크기가 다르기 때문에 IPv6 환경에서 기존의 IPv4 프로토콜의 netfilter 프레임워크를 사용하면 netfilter 프레임워크는 작동하지 않는다.

세 번째 고려사항은 IPv6 주소 출력을 위한 주소의 string 변환이다. 비정상 트래픽 제어 프레임워크에서 로그를 생성하는 기능과 사용자 인터페이스로 출력하는 기능은 프레임워크의 상태와 디버깅, 침입 트래픽을 확인하기 위해 필요하다. 이러한 기능을 지원하기 위해서 비트로 이루어진 IPv6 주소를 string으로 변환할 필요가 있다. IPv6 주소를 string으로 변환하기 위해서 비트 연산을 사용한다.

3.3 모듈별 기본 동작 및 알고리즘

이 절에서는 시스템의 각 모듈별 기본 동작 및 알고리즘을 설명한다.

3.3.1 PF 모듈

그림 1의 PF모듈은 동적으로 갱신되는 필터링 룰을 이용하여 침입으로 감지되는 트래픽을 차단한다. PF 모듈의 전체적인 흐름은 그림 4와 같다.

전체 비정상 트래픽 대응 프레임워크가 시작되면 PF 모듈은 우선 netfilter 프레임워크의 훅 포인트에 netfilter hook function을 등록시킨다. 등록한 netfilter hook function은 패킷이 하위 계층에서 전달될 때마다 호출된다. 이 netfilter hook function은 필터링 룰 셋을 검사하여 패킷의 차단 여부를 결정한다. 패킷을 차단하지 않는다면 PA 모듈로 전달한다.

3.3.2 PA 모듈

그림 1의 PA모듈은 PF 모듈로부터 패킷을 받고 이 패킷을 각 프로토콜별 메시지 큐에 넣어서 IA 모듈에서 처리하게 하고 상위 프로토콜로 패킷을 전달한다. PA 모듈은 그림 5와 같은 구성을 가진다.

PA 모듈은 패킷을 상위 프로토콜로 전송하는 과정에 추가된 모듈이기 때문에 네트워크 작업 처리 시간에 영향을 준다. 따라서 PA 모듈은 공격 탐지를 위한 메시지를 생성하여 각 메시지 큐에 넣는 작업만 하도록 하고 이 메시지를 바탕으로 침입을 탐지하는 작업은 독립적으로 동작하는 IA 모듈에서 수행한다.

3.3.3 IA 모듈

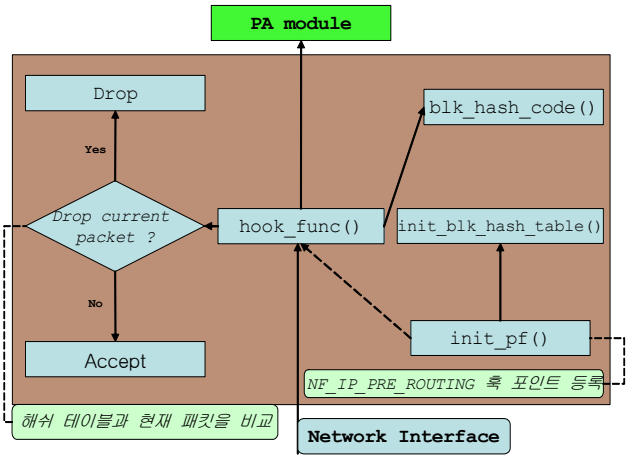
그림 1의 IA모듈은 패킷이 실제 공격인지 아닌지 탐지하는 모듈이다. 전체적인 구성은 그림 6과 같다.

IA 모듈은 커널 타이머에 등록되어 메시지 큐로부터 주기적으로 헤더 정보를 읽어온 후 등록된 필터링 룰 셋을 바탕으로 침입 여부를 판단하며, 탐지된 침입 정보를 능동적인 대응을 위해 IP 모듈로 넘겨준다. 만약 기존의 상용 침입탐지 시스템과 연동하여 탐지를 한다면 이 모듈을 거치지 않고 침입 정보를 IP 모듈로 직접 전달한다.

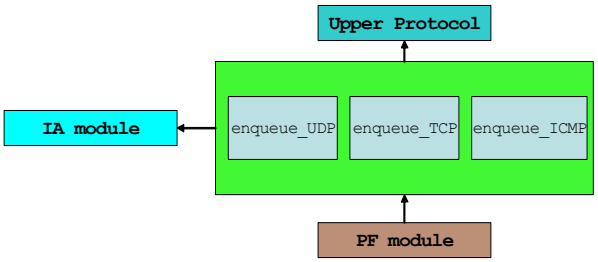
3.3.4 IP 모듈

그림1의 IP모듈은 IA 모듈로부터 공격으로 간주된 트래픽에 대한 대응을 하는 모듈이다. 전체적인 구성은 그림 7과 같다.

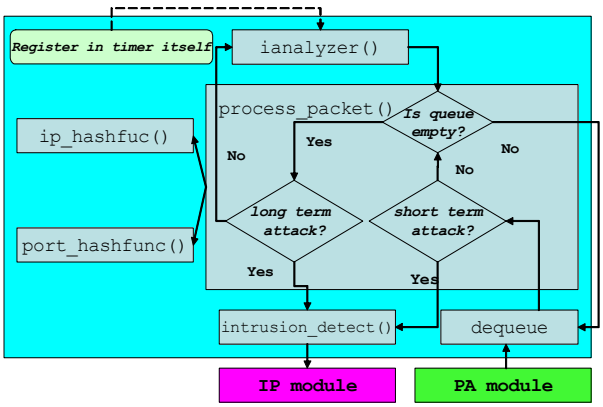
IP 모듈은 IA 모듈로부터 전달된 침입으로 간주되는 트래픽에 대응하기 위해 차단 리스트에 이 트래픽을 올려 PF 모듈에서 차단을 시키게 하거나 의심스러운 트래픽을 CBQ 메커니즘을 이용하여 트래픽이 사용하는 큐의 대역폭을 줄인다.



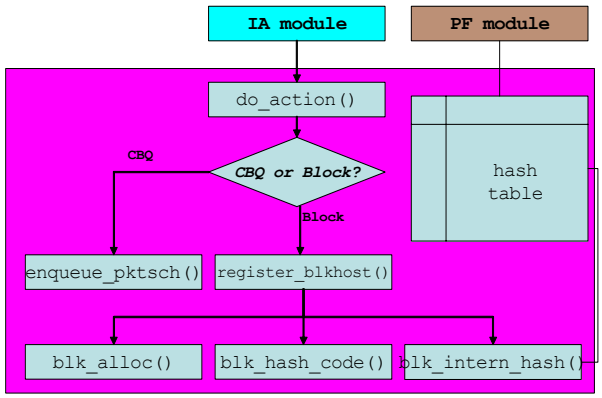
(그림 4) PF 모듈



(그림 5) PA 모듈



(그림 6) IA 모듈



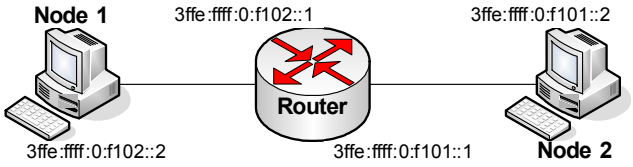
(그림 7) IP 모듈

4. 실험 및 결과

본 절에서는 제안 프레임워크를 실험하기 위해 테스트베드를 구축하고 실험 및 결과를 보인다.

4.1 테스트베드 구성

IPv6 네트워크 기반에서 제안 프레임워크를 테스트하기 위해 라우터 1대와 노드 2대를 구성한다. 리눅스 기반 IPv6 네트워크용 노드로 사용하기 위해 IPv6 네트워크 지원 옵션을 설정하고 추가적으로 Netfilter6, TC(Traffic Control), CBQ 기능을 지원하도록 커널 옵션을 설정하여 커널을 컴파일한다. 컴파일 후 ping6, traceroute6, ip도구와 같은 IPv6 네트워크 지원 네트워크 도구들을 설치하고 ip도구를 이용하여 IPv6 주소를 할당한다. 테스트베드의 구성은 그림 8과 같다.

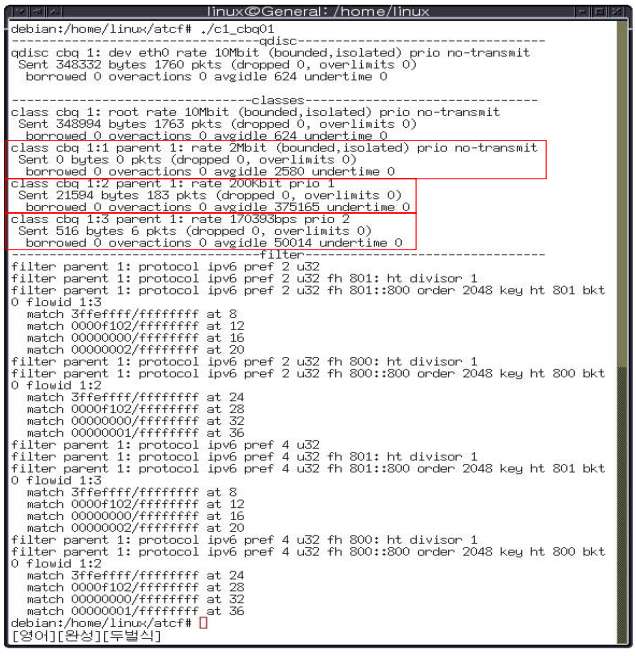


(그림 8) IPv6 네트워크 테스트베드

그림 8과 같이 라우터에 본 논문에서 제안하는 프레임워크를 설치하고 공격 노드를 노드 1로, 공격 대상 노드를 노드 2로 설정한 후 노드 1에서 노드 2로 공격 시도한다.

4.2 실험 및 결과

본 논문에서 제안하는 프레임워크는 그림 9와 같이 비정상 트래픽에 대해 CBQ 메커니즘을 이용하여 대역폭을 제어한다. 또한, 포트스캔 공격, 서비스 거부 공격, 분산 서비스 거부 공격에 대해서도 대역폭 제어 후 iptables 메커니즘을 이용하여 차단한다.



(그림 9) 단계적 대응 프레임워크에 CBQ 정책 적용

5. 기능 비교

본 절에서는 IPv6 환경을 지원하는 방화벽/침입탐지 시스템과 본 논문이 제안하는 비정상 트래픽 제어 프레임워크가 제공하는 기능을 비교한다. 표 1에서 보면 본 논문이 제안한 프레임워크는 방화벽이나 침입탐지 시스템에 비해 IPv6 네트워크에서 제공하는 IPsec 기술이 가지는 대용량의 비정상 트래픽 처리 한계를 극복하면서 CBQ 메커니즘을 사용하여 오탐지율을 줄이고 커널 레벨에서 동작하므로 시스템 자체에 성능을 높여주는 장점이 있다.

반면 Linux 시스템 상의 netfilter 프레임워크와 CBQ 메커니즘을 사용하고 커널 레벨로 동작하기 위하여 모듈로 작성하기 때문에 다른 운영체제에 포팅하기가 어렵다는 단점이 있다. 또한 CBQ 메커니즘을 이용함으로서 비정상으로 의심되는 트래픽의 대역폭을 설정하여 오탐지율을 낮추지만 적절한 대역폭을 설정하기 어려운 단점이 있다.

구분	방화벽/침입탐지 시스템	제안 프레임워크
장점	● 플랫폼에 독립적 ● 다른 보안 시스템과 공조 가능 ● 보안정책 수립용이	● 오탐지율을 낮춤 ● IPsec 기술이 적용된 대용량의 비정상 트래픽 제어 용이 ● 불특정 다수의 사용자를 대상으로하는 네트워크에 적용 용이 ● 커널 레벨에서 동작하므로 전체적인 성능향상
단점	● 실시간으로 대용량의 비정상 트래픽에 대응하지 못함 ● 오탐지율이 증가함 ● IPsec 기술이 적용된 패킷 처리 어려움	● Linux 이외의 다른 운영체제로 포팅이 어려움 ● CBQ 메커니즘을 위한 최적의 대역폭 설정의 어려움

(표 1) IPv6 환경에서 방화벽/침입탐지 시스템과 제안 프레임워크 기능비교

6. 결 론

본 논문에서는 IPv6 네트워크에 적용할 수 있는 비정상 트래픽 제어 프레임워크를 제시하였다. IPv6 프로토콜에서 제공하는 IPsec은 인증과 신뢰성, 무결성 등의 기능을 제공하지만 대용량의 비정상 트래픽에 의한 공격에 취약하다. 이러한 IPsec 기술이 가지는 한계를 극복하기 위해 본 논문이 제시한 프레임워크는 netfilter6 프레임워크를 사용하여 IPv6 환경에서 대용량의 비정상 트래픽에 실시간으로 대응하고, 오탐지율을 줄이기 위해 CBQ 메커니즘을 이용하여 의심스러운 트래픽의 대역폭을 줄이는 메커니즘을 제시하였다. IPv6 환경을 지원하기 위해 IPv6 프로토콜 주소 구조체를 사용하는 것을 제안하였고 로깅 기능과 사용자 인터페이스를 위해서 IPv6 주소를 string으로 변환하는 기술을 제안하였다.

향후 연구 내용으로는 실제 프레임워크를 구현하고 IPv6 네트워크 환경에서 성능을 평가를 수행하여 실제 이 프레임워크를 평가하고자 한다.

참고문헌

[1] S. Deering and B. Hinden , “Internet Protocol, Version 6 (IPv6) Specification,” RFC 2460, Dec. 1998.
[2] Kent, S. and R. Atkinson, "Security Architecture for Internet Protocol," RFC 2401, Nov. 1998.
[3] Rocky KC Chang, "Defending Against Flooding-Based, Distributed Denial-of-Service Attacks : A Tutorial," IEEE Communications Magazine, Oct. 2002.
[4] 조은경, 고광선, 이태근, 강용혁, 엄영익, “리눅스 Netfilter 프레임워크와 CBQ 라우팅 기능을 이용한 비정상 트래픽 제어 시스템 설계,” 정보보호학회논문지, 한국정보보호학회, Vol. 13, No. 6, May 2003.
[5] Alessardo Rubini and Jonathan Corbet, Linux Device Driver, O'Relly, 2nd Edition, 2002.
[6] Klaus Wehrle, et al, The Linux Network Architecture, Prentice Hall, 2005.