논문 2005-42CI-4-3

안전한 액티브 네트워크 구조에 관한 연구

(Study on a Secure Active network Architecture)

홍 성 식*, 한 인 성*, 유 황 빈*

(Sung Sik Hong, In Sung Han, and Hwang Bin Ryou)

요 약

기존의 수동적인 네트워크는 단순히 데이터의 저장과 전송기능만을 가지고 있다. 이와 달리 액티브 네트워크는 네트워크 전송로상에서 전달중인 패킷에 대한 연산작업이 가능한 네트워크 패러다임이 1990년대 소개되었다. 하지만, 이와 같은 네트워크 패킷에 대한 능동적인 처리가 가능하다는 특징은 보안에 관한 측면에서는 보다 많은 보안상의 위협가능성을 고려해야만 한다.

본 논문에서는 이와 같이 능동적인 처리가 가능한 액티브 네트워크 환경의 취약점을 보안하고자 패킷 분리 방식을 이용한 안전한 액티브 네트워크 구조를 제안한다. 제안하고자 하는 시스템은 액티브 노드의 관리와 서비스의 이용자에 대한 관리가 가능한 모델로 설계하며, 패킷의 암호화와 세션을 이용한 패킷 전송을 통해 보안성을 향상시키고자 한다.

Abstract

The existing passive networks have the only data-storing and transmission functions. On the other hand, the active network which can do operation jobs on the transmitting packets was introduced at 1990's. However, the advantages of activating processing are obviously more complex than traditional networks and raise considerable security issues.

In this paper, we propose the safer structure in Active Networks that is based on the discrete approach which resolves the weak point of the Active Network. The proposed system provides the node management and user management in the Active Networks, and improves the security of packet transmission with packet cryptography and the session.

Keywords : active network, network architecture, discrete approach, security, cryptography

I. 서 론

기존의 패시브 네트워크 구조는 새로운 기술에 대한 수용과 보급이 구조적인 특성으로 인해 많은 시간 지연이 발생하고 있는 실정이다. 그러므로 새로운 네트워크 서비스를 요구하는 네트워크 이용자들의 요구를 만족시키에는 많은 어려움들이 있다. 또한 새로운 기술들에 대한 표준이 제정되기까지 재정적, 시간적 지연이 발생하게 되고 이로 인해 많은 네트워크 자원과 시간적 손실이 발생하고 있는 실정이다[3].

DARPA에서는 이러한 기존의 패시브 네트워크 구조의 문제점을 보완할 수 있는 새로운 패러다임으로 액티브 네트워크를 제안하였다. 액티브 네트워크는 단순히 패킷의 전송만을 처리하는 패시브 네트워크에 프로그램 처리를 할 수 있는 액티브 노드(프로그래머블 라우터나 스위치 장비)를 배치해 사용자들의 요구에 따른 패킷의 연산과 처리를 할 수 있도록 한 새로운 차세대 네트워크 구조이다. 액티브 네트워크 환경에서는 새로운 서비스를 보다 신속하게 배치할 수 있고 고성능의 네트워크 장비들을 활용할 수 있어 자원들을 적절하고 효율적으로 활용하는 것이 가능하다.

그러나 액티브 네트워크내에서 전송되는 패킷에 포함된 프로그램 코드를 액티브 노드에서 실행할 수 있다는 장점이 오히려 보안상의 커다란 위협요소로 동작할 수 있는 위험요소를 내재하고 있다. 더욱 복잡해지는 네트워크 구조에서 전송중인 패킷을 악의적인 목적으로 이요하여 네트워크를 구성하는 노드들의 붕괴, 변경,노

* 정회원, 광운대학교 컴퓨터소프트웨어학과
(Dept of Computer Science & Engineering, Kwang-woon University)

※ 이 논문은 2002년도 광운대학교 연구년에 의하여 연구되었음

접수일자: 2005년3월29일, 수정완료일: 2005년6월27일

드 리소스들에 대한 서비스 거부 공격 및 비밀정보 도청 등과 같은 예를 들어볼 수 있다. 그러므로, 액티브 네트워크 구조가 갖는 많은 장점들을 활용함에 있어 먼저 보안적인 위협요소들을 우선적으로 해결하는 것이 시급한 사안이다[4]. 이같은 보안상의 문제점들을 보안하기위해 본 논문에서는 액티브 네트워크 구조중 패킷 분리방식을 이용한 액티브 네트워크 구조에서 사용자의 서비스 요구를 관리하는 보안 시스템을 제안한다.

제안하고자 하는 시스템은 액티브 노드에서 동작할 수 있는 프로그램 코드들을 배치하는 사용자에 대한 관리를 한다. 또한 액티브 노드 자체에 대한 인증관리를 기본으로, 각각의 액티브 노드들 사이에서 전송되는 패킷에 대한 비밀성을 보장하기 위해 패킷의 암호화와 세션 기법을 이용해 전송된 패킷에 대한 안전성을 보장한다. 이러한 보안 기술등을 이용해 안전한 프로그램 코드의 전송 및 배치방안을 개선하고자 한다. 제안된 액티브 노드 보안 시스템 모델의 설계와 구현 과정을 통해 기존의 액티브 네트워크 구조와 비교해 보다 향상된 안전성을 제공할 수 있고, 이로 인한 효율성의 반감을 최소화하였다[6,10].

본 연구논문의 구성은 다음과 같다. 서론에서는 논문에 대한 필요성과 이에 대한 연구방향을 제시하고, Ⅱ장에서는 기존에 선행된 액티브 네트워크에 관한 관련 연구와 구성방식에 대해 설명한다. Ⅲ장에서는 본 논문이 제안하고자 하는 안전한 액티브 네트워크 구조에 대한 소개와 이를 활용한 효과적인 액티브 네트워크 구조의 보호방법에 대해 살펴본다. Ⅳ장에서는 본 논문이 제안하는 액티브 네트워크 구조의 설계와 구현에 관해 설명하고, Ⅴ장에서 본 논문에 대한 연구결과를 보이며, 마지막으로 Ⅵ장에서는 본 논문의 결론과 향후 연구방향에 대해 기술하였다.

Ⅱ. 기존 연구

1. 액티브 네트워크 구성방식

액티브 네트워크는 기존의 패시브 네트워크 구조와는 다른 네트워크 구조로서, 네트워크 노드에서 사용자의 요구에 알맞은 동작을 수행시키고자 하는 기본적인 목적을 만족시키기 위해 다양한 연구 접근 방법들이 개발되었다. 또한 네트워크 노드에서 사용자의 요구를 만족시키는 동작을 수행하기 위해 이러한 노드들에 대한 인증이나 실행환경에 대한 일치성을 제공하기 위한 연구와 실행환경에서 사용되기 위한 특별한 프로그램 언어에 대한 연구들이 선행되었다.

위와 같이 다양한 연구관점과 목적을 만족시킬 수 있는 많은 연구방법들이 수행되었으나, 본 논문에서는 노드에서 사용자의 욕구를 만족시키기 위해 전송되는 패킷에 포함된 프로그램 코드와 데이터를 안전하게 전송하도록 하는데 초점을 두고 있다.

ANTS(Active Network Transfer System)은 전송하는 패킷내에 프로그램 코드를 포함시켜 지정된 액티브 노드에 필요한 기능을 설치할 수 있도록 하는 가장 초기의 연구로써, 액티브 네트워크에 대한 구조와 종합적인 연구결과들을 생성하였다. 또한 액티브 네트워크 구조에서 안전성과 보안적인 측면에서 유연하게 프로그램 처리가 가능하도록 하는 기능을 강화한 네트워크 장비가 연구되었다. 이와 함께 현실적으로 액티브 네트워크 기반 구조를 형성하는데 있어 많은 문제점들을 연구하고, 가상의 액티브 네트워크 구조를 구성한 ABone (Active Network Backbone)과 액티브 패킷(액티브 네트워크에서 사용할 패킷) 구조를 설계하고 다양한 실행환경을 지원하도록 하였다. 이러한 기존의 연구들로 인해 액티브 네트워크를 보다 다양하고 많은 분야에서 연구할 수 있는 초석을 마련하였다.

기존의 액티브 네트워크 구성은 패킷의 전달방식에 따라 구분할 수 있는데, 크게 프로그램 코드와 데이터를 분리하여 처리하는 패킷 분리방식과 프로그램 코드와 데이터를 함께 액티브 패킷이라는 구조로 통합하여 전달하는 캡슐방식으로 구분할 수 이다.

패킷 분리방식은 패킷을 처리하는 프로그램 코드와 프로그램으로 처리할 데이터를 분리하여 전송하는 방식을 의미한다.

캡슐방식은 액티브 노드에 프로그램 코드를 저장하지 않고 액티브 노드 사용자가 프로그램 코드와 데이터를 포함한 액티브 패킷(캡슐)을 생성하여 액티브 네트워크로 전송하는 방식을 의미한다. 액티브 패킷을 수신한 액티브 노드는 수신한 액티브 패킷을 프로그램 코드와 데이터로 분리한다. 이러한 분리과정 후 분리된 프로그램 코드를 액티브 노드의 실행환경에 적재시키고, 적재된 프로그램 코드를 이용해 분리된 데이터를 액티브 노드의 실행환경에 적재된 프로그램 코드에의해 처리하게 된다. 마지막으로 프로그램 코드에 의해 처리된 데이터의 결과와 프로그램 코드는 액티브 패킷으로 다시 캡슐화되어 이웃 액티브 노드로 전달하게 된다.

MIT 대학에서 수행중인 ANTS 프로젝트와 펜실베니아 대학의 PLANet 등이 이러한 연구방식을 적용하

고 있다. 하지만, 위와 같은 캡슐방식의 문제점은 전달해야 할 프로그램 코드의 양이 큰 경우 네트워크상에서 트래픽 오버헤드 문제와 패킷 분실로 인한 재전송 문제 등의 단점을 내포하고 있어 그 효용성이 저하될 수 있다.

패킷 분리방식은 액티브 네트워크 구조에서 수행될 프로그램 코드와 데이터를 분리하여 전달하는 방식이다. 이는 먼저 수행할 프로그램 코드들을 액티브 노드에 설치하는 네트워크 구조이다. 액티브 노드 사용자는 전송할 패킷에 기록된 프로그램 코드 식별번호와 데이터를 포함시켜 데이터를 처리할 수 있는 프로그램이 설치된 액티브 노드로 전송한다. 액티브 패킷을 수신한 액티브 노드는 프로그램 코드 식별자를 이용해 식별번호를 확인하고 식별번호와 일치하는 프로그램 코드를 액티브 노드에에 적재시켜 데이터를 처리한다. 마지막으로 처리된 데이터를 액티브 패킷으로 재 생성하여 이웃노드로 전달한다. 이러한 패킷 분리방식을 이용한 기존의 연구로 ActiveIP와 SwitchWare가 선행 연구되었다. 패킷 분리 방식은 이미 이식되어 있는 프로그램 코드에 대해서만 적용가능하고 오직 네트워크 관리자만이 프로그램 코드를 추가시킬 수 있기 때문에 액티브 노드를 이용하려는 일반 호스트들은 그들이 원하는 새로운 프로그램을 설치하는 것이 불가능하다. 본 논문에서는 액티브 네트워크에 대한 연구방식 중 패킷 분류방식을 기반으로 하여 이와 같은 일반 호스트들이 액티브 노드를 이용할 수 있는 네트워크 구조를 제안한다[1,2,8,9].

2. 액티브 네트워크 구성방식

액티브 네트워크 환경에서 기본적인 보안 서비스를 제공하기 위해서 인증(Authentication), 권한 부여(Authorization), 무결성(Integrity) 등의 문제들에 대한 해결책을 제시하고 신뢰성 있는 모델을 제공하여야 한다.

본 논문에서 바탕으로한 패킷 분리방식의 액티브 네트워크 구조에서는 프로그램 코드 전송자에 대한 인증과 프로그램 코드 자체에 대한 비밀성 및 무결성 검증은 반드시 필요한 보안요소이다. 액티브 노드에 전송된 프로그램 코드가 악의적인 목적으로 변경되거나 수행중에 문제발생의 소지가 있다면 이는 예기치 않은 실행 오류로 인해 전체 액티브 노드들에 대한 성능 저하뿐만 아니라 보다 큰 보안상의 문제들을 일으킬 수도 있다.

또한, 프로그램 코드에 대한 인증이 이루어지지 않을 경우 악의적인 목적을 가진 공격자가 프로그램 코드들을 위.변조하여 네트워크 전체에 악영향을 미치게 되는 잠재적인 위협요소로 발전할 수 있다.

현재 액티브 네트워크 보안에 관한 SANE, Seraphim, PLAN, Safetynet 등의 프로젝트가 계속 수행되고 있지만 근본적인 액티브 노드에 대한 안전성을 보장하지 못하고 있는 실정이다. 이에 액티브 네트워크에 대한 보안 취약성을 보완할 수 있는 새로운 보안체계가 반드시 필요하다[4].

Ⅲ. 시스템 제안

기존의 선행연구들을 살펴본 바와 같이 액티브 네트워크 환경에서 인증, 권한부여, 무결성 등의 기본적인 보안 문제에 대한 해결책을 제시할 수 있는 신뢰할 수 있는 보안 모델의 제공은 반드시 필요하다. 만약 기본적인 보안 문제들이 무시된다면, 프로그램 가능한 액티브 네트워크 노드들은 전체 네트워크의 성능저하 뿐만 아니라 개인의 프라이버시 침해 및 네트워크의 혼란을 초해함은 물론 보다 큰 보안 위험성을 갖게 된다.

이러한 보안 위험성들을 해결하기 위해서는 패킷 분리 방식을 기반으로 하는 액티브 네트워크 구조에서 액티브 노드의 사용자들을 인증해야 한다. 액티브 노드의 사용자들을 오류없이 인증함으로써 악의적인 프로그램 코드를 전송하는 공격자의 접근을 제한할 수 있고, 전송되는 프로그램 코드의 위.변조를 막을 수 있다. 또한 프로그램 코드의 서비스 이용자(액티브 클라이언트)들이 자주 사용하는 프로그램 코드들에 대한 관리를 통해 액티브 노드의 프로그램 재 설치로 인한 수행성능 저하를 감소시킬 수 있을 것이다.

본 논문에서 제안하고자 하는 안전한 액티브 네트워크 구조는 그림 1과 같다.

제안하는 액티브 네트워크 구조는 기본적으로 액티브 노드에 대한 인증과 액티브 노드에 설치되는 프로그

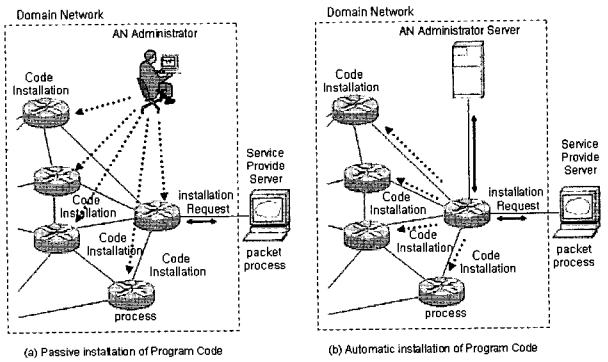


그림 1. 제안된 안전한 액티브 네트워크 구조
Fig. 1. Proposed Safe Active Network Architecture.

램 코드들을 안전하게 전송하는 문제를 해결하고자 하는 것이다. 제안된 구조는 액티브 노드의 인증 문제를 해결하기 위해 우선적으로 신뢰할 수 있는 인증기관으로부터 액티브 노드를 관리하는 액티브 노드 관리 서버가 인증서를 발급해주는 것을 가정한다. 이는 기존의 인터넷 서비스에서 상호 신뢰를 위해 인증 기관을 이용한 인증기법인 PKI와 유사한 기반구조이다.

네트워크 도메인 내에서 신뢰할 수 있는 액티브 노드 관리 서버는 액티브 네트워크 구조내에 존재하는 서비스 이용자들을 인증하고 관리한다. 또한 각각의 네트워크 도메인내에 신뢰할 수 있는 액티브 노드 관리 서버들은 서로간의 상호인증이 가능하다.

액티브 노드 관리 서버들은 서비스 이용자들을 인증함은 물론 서비스 이용자가 이용하려는 프로그램 코드들에 대해서도 인증과 관리를 수행한다. 이를 위해 먼저 서비스 이용자들은 자신이 이용하려는 프로그램 코드와 서비스 이용자 대한 인증 서비스를 신청하고 이를 승인받게 된다. 인증된 프로그램 코드에 대해서는 액티브 노드 관리 서버에 등록되고 필요에 따라 액티브 노드에 설치되어 실행이 가능하다.

기존에 연구된 패킷 분리방식의 단점은 서비스 이용자가 사용하려는 프로그램 코드를 액티브 노드에 적재하는 것이 네트워크 관리자만이 관리 및 설치가 가능하다는 것이다. 이러한 제한 점을 보완하기 위해 모든 서비스 이용자들이 서비스를 요청하면, 액티브 노드 관리 서버는 서비스 이용자들을 인증하고 서비스를 이용하려는 프로그램 코드를 액티브 노드 관리 서버에 등록한다. 등록된 프로그램 코드에 대한 보안 위협요소들에 대한 점검이 정당한 경우에 액티브 노드에 프로그램 코드를 설치해 서비스 이용자가 요청하는 연산을 수행한다. 프로그램 코드를 요청에 따라 인증코드만을 사용하여 프로그램이 수행토록 함으로써 안전하게 액티브 노드의 활용성을 높일 수 있다. 또한 액티브 노드의 관리를 통해 액티브 노드 관리 서버에 등록되어 있는 액티브 노드로의 데이터 전송에 암호화 및 인증 뿐만 아니라 프로그램 코드의 분석을 통한 위.변조 방지 기능을 갖추고 있다.

Ⅲ. 시스템 설계

앞서 제안한 바와 같이 기존의 패킷 분리 방식을 이용한 액티브 네트워크 구조에 대한 보안 및 효율성 향상을 위해 액티브 노드 관리 서버를 다음과 같이 기능적으로 모듈화하여 설계하였다.

액티브 네트워크 구조에 대한 설계는 액티브 노드 관리 서버와 액티브 노드들을 동작하도록 하는 액티브 노드 에이전트의 쌍으로 설계하고자 한다. 액티브 노드 관리 서버에서 액티브 노드의 요청에 따라 프로그램 코드를 등록하기 위한 사전 절차들을 처리하고 액티브 노드 에이전트에게 필요한 정보들을 제공하며 이를 관리하도록 구성하였다.

액티브 노드의 요청을 받은 액티브 노드 관리 서버는 액티브 노드와 통신에 사용할 키쌍을 생성하여 분배한다. 분배된 키 정보를 이용해 액티브 노드에게 액티브 노드 에이전트를 전달하고 액티브 노드는 전달받은 액티브 노드 에이전트를 이용하여 액티브 노드와 상호 동작하게 된다. 이같은 과정을 통해 안전하고 효율적인 액티브 네트워크 구조가 이루어지게 된다.

액티브 노드 관리 서버부분은 기능에 따른 세부적인 모듈로 나누어 볼 수 있다. 기능별 세부 모듈은 액티브 노드 키 관리모듈, 액티브 노드 제어 모듈, 프로그램 코드 관리 모듈, 액티브 노드 실행환경 정보저장소와 같이 크게 4개의 모듈로 구성된다. 4개의 기능별 모듈은 그 역할과 기능에 따라서 세분화된 형태로 구분되어 자신이 수행해야 할 역할을 담당하고 있다.

1. 액티브 노드 관리 서버

가. 액티브 노드 키 관리 모듈

액티브 노드·키 관리 모듈은 액티브 노드에 프로그램 코드 등록요청을 받아 노드의 IP 정보를 이용해 공개키와 개인키 쌍을 생성한다. 이러한 과정에서 생성된 개인키를 액티브 노드로 전달해 생성된 공개키를 노드의

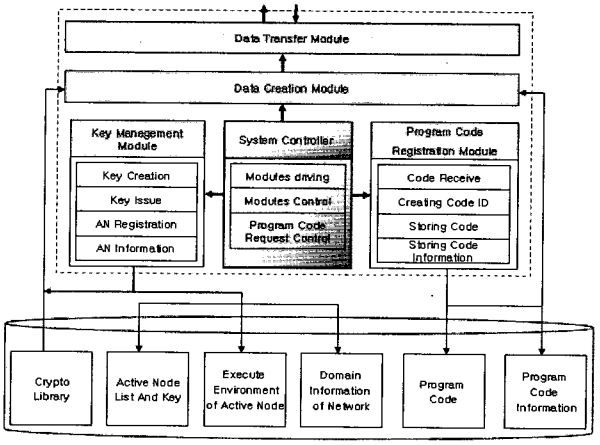


그림 2. 액티브 노드관리 시스템 구조
Fig. 2. Architecture of Active Node Administration System.

실행환경 정보 저장소에 저장한다.

네트워크 도메인 내에 관리해야 할 액티브 노드가 많을 경우 액티브 노드 키 관리자는 노드의 공개키들만을 저장하고 관리하는 모듈과 공개키 그리고 개인키 쌍에 대한 정보를 저장 관리 모듈로 구체화 시킬 수 있다.

나. 액티브 노드 제어 모듈

액티브 노드 제어모듈의 주요 동작은 액티브 노드 관리 서버를 구성하는 각각의 모듈과 액티브 노드 에이전트간의 상호동작의 수행을 통해 발생하는 여러 가지 문제들에 대한 조정과 관리를 한다.

구체적으로, 기능에 따른 모듈을 세분화하면, 액티브 노드와의 데이터 송.수신을 위한 연결세션 설정, 액티브 패킷 전송, 암호화 처리작업, 프로그램 코드에 대한 암호화 및 비밀키 생성 등의 작업을 수행하는 데이터 전송 모듈 부분과 패킷의 해데 생성 및 프로그램 코드의 수집, 전송할 데이터에 대한 캡슐화를 수행하는 데이터 생성 모듈로 구분할 수 있다.

다. 프로그램 코드 관리 모듈

프로그램 코드 관리 모듈의 처리기능은 액티브 노드로 액티브 패킷을 전송해 액티브 패킷의 데이터를 연산처리를 할 수 있는 프로그램 코드의 저장과 관리를 담당한다. 프로그램 코드 제공자의 요청에 따라 사용 가능여부를 확인하고 정당한 사용자라면 액티브 노드에 전송하여 설치한 후 사용하도록 한다.

이를 위해서는 프로그램 코드 제공자가 요구하는 프로그램 코드들을 수집하여 액티브 노드 관리 서버에 저장하도록 하는 모듈, 프로그램 코드 제공자를 인증하는 인증하는 모듈, 저장된 프로그램 코드가 수행되기 위한 환경정보 관리모듈, 프로그램 코드의 특별한 세부사항들을 저장하기 위한 모듈로 구체화하여 설계하였다.

프로그램 제공자로부터 수신된 프로그램 코드를 정보저장소에 저장하고 이를 효과적으로 관리하기 위해 10자리의 직렬번호로 구성된 프로그램 코드 저장 키 정보구조를 생성해 각각의 프로그램 코드들을 관리하도록 하였다. 이러한 과정을 통해 각각의 프로그램 코드를 이용하는 사용자의 요청에 대해 검색, 전송 및 관리에 있어서 효율성을 높일 수 있다.

라. 액티브 노드 실행환경 정보저장소

액티브 노드 실행환경 정보 저장소가 수행하는 작업을 간단하게 다음과 같이 설명할 수 있다. 액티브 노드 실행환경 정보저장소가 수행하는 작업은 프로그램 코드의 이용자나 프로그램코드 제공자에 의해 제공되는 프로그램 코드와 10자리 직렬번호로 구성된 프로그램 코드 저장키, 데이터의 전송에 필요한 공개키, 세션 설정 후에 사용되는 비밀키 등과 같은 정보들을 각각의 액티브 노드 관리 서버의 세부모듈에서 필요로 하는 정보들을 각각의 디렉토리로 구성하여 데이터베이스로 저장하고 관리한다.

또한, 수행요청을 받은 프로그램 코드에 대한 실행환경 정보도 데이터베이스화하여 같이 저장하여 관리함으로써 액티브 노드에 설치된 액티브노드 에이전트와의 통신에 효율성을 높일 수 있다.

2. 액티브노드 에이전트

액티브노드 에이전트란 액티브 노드에 설치되어 액티브 노드 관리 서버와의 통신을 수행한다. 주요 기능은 액티브 노드의 상태정보 수집과 수집된 정보를 액티브 노드 관리 서버로 전달하는 것이다. 이를 통해 액티브 노드 관리 서버는 액티브 노드의 실행환경을 확인할 수 있다.

또한, 액티브 노드 관리 서버와 비밀성이 보장되는 통신을 수행하기 위해 생성된 비밀키쌍을 전송받아 보안상의 위협요소로부터 안전하게 저장하여 관리하는 기능도 갖는다. 암호화된 프로그램 코드 및 데이터를 송.수신하기 위해서 세션키 교환기능도 수행한다.

V. 모의실험 결과 및 고찰

본 장에서는 실험 모델을 설명하고 실험 결과를 분석한다. 본 실험에서는 액티브 패킷의 실행을 위해 제안한 방법들을 이용해 액티브 노드에 프로그램코드를 추가함으로써 액티브 노드의 성능에 어떤 영향을 미치는지 성능을 평가하고자 한다. 그리고 이러한 성능평가를 위해 액티브 노드에서 클라이언트의 요청을 처리할 수 있는 프로그램 코드를 적용한 액티브 노드를 위치한 경우 WWWServer에서의 부하율 및 서비스 처리 성능을 일반노드를 적용한 경우와 비교 평가한다.

1. 실험환경

액티브 노드 관리자 서버로부터 입력된 실험 프로그램 코드를 내장한 액티브 노드의 실험환경은 GT-ITM으로 생성한 트랜짓-스텁(Transit-Stub) 구조로 현재의 인터넷 환경과 유사함을 보인다. 또한 실험에 이용된

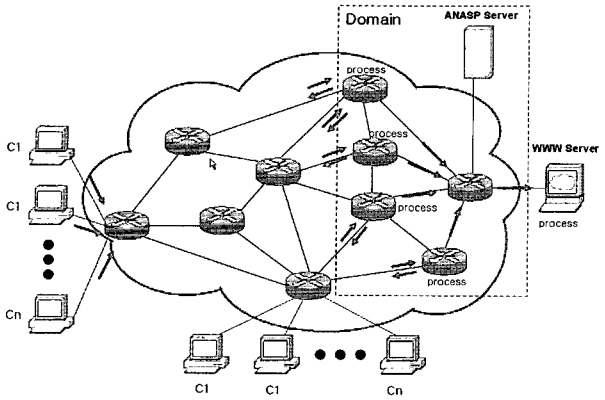


그림 3. 실험을 위한 액티브 네트워크 토폴로지
Fig. 3. An Experimental Active Network Topology.

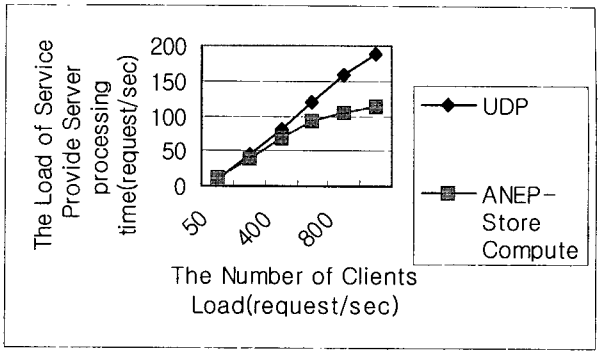


그림 4. 클라이언트 수의 증가에 따른 Service Provide Server 처리 량
Fig. 4. Protocol latency of Service Provide Server Load.

도구로는 UCB(University of California, Berkeley)의 LBNL(Lawrence Berkeley National Laboratory)에서 개발한 ns2를 사용하였다. 액티브 노드의 트래픽은 액티브 노드를 이용한 네트워크 지연시간과 클라이언트의 홉 카운트를 TCP 와 UDP 그리고 ANEP 패킷을 이용하여 성능측정을 하였다.

그림 3은 본 논문의 연구를 위해 구성한 네트워크 토폴로지를 보여주고 있다.

서로 다른 도메인에 존재하는 서버와 클라이언트로 구성된 네트워크를 구성한다. 이를 위해서 다른 도메인에 존재하는 라우터간의 연결을 가정한다. Service Provide Server의 요청으로 액티브 노드 관리서버는 관리 도메인내의 액티브 노드로 캐쉬기능을 처리하는 프로그램 코드를 설치한다. 클라이언트들은 Service Provide Server로 서비스 요청 메시지를 전송하게 되면 프로그램 코드가 설치된 액티브 노드와 Service Provide Server는 클라이언트의 요구를 처리한다. 캐쉬기능을 갖는 액티브 네트워크와 비교하기 위한 모델로 TCP와 UDP를 채택하였다. TCP는 인터넷에서 신뢰성 전송을 담당하는 대표적인 전송 프로토콜이지만 많은 기능으로 인하여 부하가 크다. UDP는 ANEP 패킷을 캡슐화하지 않은 경우에 해당한다. 단순히 전송 기능만 제공하므로 신뢰성을 보장하지 못하는 반면 부하가 작다. 성능비교를 공정하게 하기 위하여 같은 양의 데이터를 같은 패킷 크기로 전송하였다. 비교를 위한 트래픽은 모두 수신자와 송신자를 임의로 20개씩 택하여 전송하였다.

2. 모의실험 결과 및 분석

본 실험에서는 백그라운드 트래픽이 없을 때를 가정하여 실험을 하였다. 이는 액티브 패킷의 손실이 없는 상황에서 정상적으로 작동하는지 시험할 수 있고 손실 복구 능력과는 별도로 액티브 노드의 프로세스의 성능을 테스트할 수 있다. 이러한 백그라운드 트래픽이 없을 때 액티브 네트워크의 최고의 성능을 발휘할 수 있다. 혼잡으로 인하여 패킷의 손실이 발생하지 않기 때문에 순수하게 액티브 네트워크의 성능 테스트를 할 수 있다.

본 실험을 위한 프로그램 코드는 클라이언트와 Service Provide Server사이의 중간 액티브 노드에 설정되어 들어오는 패킷의 요청에 일치하는 데이터가 없는 경우 Service Provide Server로부터 전송되는 결과를 액티브 노드에 저장하고 요청에 일치하는 캐쉬데이터가 있는 경우 데이터를 포함한 패킷을 클라이언트로 전송하는 캐쉬데이터처리기능을 수행한다. 이미 클라이언트 사이트 캐쉬의 경우는 널리 사용되고 있는 기술이지만, 본 논문에서는 이러한 프로그램 코드를 이용하여 액티브 노드를 적용한 네트워크와 일반 네트워크 추가적인의 성능을 테스트 하였다.

그림 4은 클라이언트 수에 따른 Service Provide Server에서의 처리시간을 측정한 결과이다. Service Provide Server는 병목현상이 발생할 수 있으며, 클라이언트의 수가 증가에 비례하여 Service Provide Server의 전송지연시간을 측정한 결과를 그래프로 나타낸것이다. 모의실험 결과에서 보는바와 같이 Service Provide Server의 시간이 감소하는 현상을 볼 수 있다. 이러한 결과는 많은 중복결과가 각각의 다른 클라이언트들에 의해 발생되기 때문이다.

본 실험을 통해 UDP 프로토콜을 이용한 일반 네트워크와 캐쉬기능을 처리하는 프로그램코드를 갖는 액티브 네트워크를 비교한 결과를 보였다. 하지만 종단 액티브 노드에 캐쉬기능을 갖는 액티브 네트워크가 가장

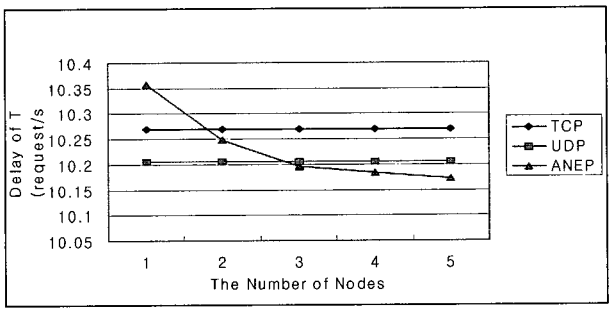


그림 5. 액티브 노드 수에 따른 프로토콜별 처리성능
Fig. 5. Client request-response time.

효과적이라고 명확히 할 수는 없다. 본 실험은 단지 아주 단순한 캐싱기능에 대한 처리를 하였기 때문이다. 실험의 명확성을 위해서 일반 네트워크에서의 적용이 필요하다.

그림 5는 액티브 노드의 수를 증가시키면서 Service Provide Server로 서비스 요청 패킷들을 보냈을 때의 클라이언트 요청응답 지연시간을 나타낸 결과이다. 실험에 적용한 클라이언트의 수는 500개를 기준으로 하였다. 프로토콜은 종단간의 전송만을 담당하는 TCP와 UDP 그리고 실험을 위해 액티브 노드에서 패킷을 확인하고 연산처리 후 캐쉬 데이터를 포함하여 전송 할 수 있는 ANEP를 비교하였다.

그림 5에서 알 수 있는바와 같이 TCP와 UDP 프로토콜은 중간의 액티브 노드의 수와 상관없이 일정한 값을 갖는다. 하지만 ANEP 패킷은 TCP 와 UDP 프로토콜에 비해 클라이언트 요청응답 지연시간이 감소하고 있는 현상을 볼 수 있다. 이러한 결과는 액티브 노드의 수가 증가할수록 ANEP패킷의 처리율이 짧아 진다는 것을 의미한다. 즉, 액티브 노드의 수가 증가할 수 록 패킷의 처리 결과가 돌아오는데 걸리는 시간이 줄어들기 때문이다.

VI. 결 론

액티브 네트워크는 전송중인 액티브 패킷의 프로그램 코드를 액티브 노드에서 실행할 수 있으며, 네트워크로 전송된 프로그램 코드의 실행 결과에 따라 액티브 노드의 상태를 변경할 수 있어, 패킷의 전송 기능만을 수행하는 패시브 네트워크에 비해 더욱 복잡한 네트워크 상태를 갖게 된다. 이로 인해 보다 많은 보안상의 위협과 공격이 훨씬 쉽고 다양한 방법이 가능하다.

본 논문에서는 악의적인 목적으로 액티브 노드의 취약성을 이용해 악의적인 프로그램 코드가 설치되거나, 전송중인 프로그램 코드의 위·변조를 통해 네트워크 전체를 위협할 수 있는 보안상의 취약점을 해결하기 위해, 관리 도메인 내에 존재하는 액티브 노드들을 효율적으로 관리하고 액티브 노드로 전송되는 패킷의 위·변조를 막을 수 있도록 암호화를 통해 프로그램 코드를 전송하는 액티브 노드 관리 시스템을 설계 및 구현하였다.

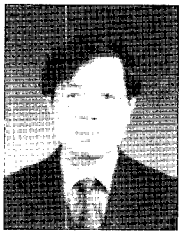
또한 액티브 노드 관리 시스템을 이용해 캐쉬 프로그램 코드가 설치된 액티브 노드를 이용하여 프로그램 코드의 설치 지연시간과 프로토콜별 처리 지연시간을 실험하고 이를 측정하여 액티브 네트워크의 성능을 분석하였다.

참 고 문 헌

[1] K. Calvert, et.al., "Direction in Active Networks", IEEE Comm. Mag., Oct. 1998.
[2] Danny Raz and Yuval Shavitt, "An Active Network Approach to Efficient Network Management", WAN'99, 1999.
[3] A. B. Kulkarni,". Implementation of a Prototype Active Network.", In OPENARCH '98, 1998.
[4] AN Security Working Group, "security Architecture for Active Nets", Nov, 2001.
[5] Konstantinos Psounis, "Active Networks: Applications, Security, Safety and Architectures", IEEE Communications Surveys, First Quarter 1999.
[6] R. H. Campbell, et al., "Seraphim: Dynamic Interoperable Security Architecture for Active Networks", IEEE OPENARCH 2000, Tel-Aviv, Israel, Mar. 2000.
[7] D. Tennenhouse and D. Wetherall, "Towards an Active Network Architecture," Computer Communication Review 26(2), April 1996.
[8] D. Tennenhouse et al, "A Survey of Active Network Research," IEEE Communications Magazine, January 1997.
[9] D. Wetherall and U. Legedza and J. Guttag, "Introducing new internet services: Why and how", IEEE Network Magazine, 1998.
[10] D. J. Wetherall, "Service Introduction in an Active Network", Ph.D. Thesis Submitted to the Department of Electrical Engineering and Computer Science, M.I.T., Feb. 1999.

저 자 소 개

홍 성 식(정회원)
1989년 광운대학교 전자계산학과 학사졸업
1992년 광운대학교 전자계산학과 석사졸업.
1994년 혜전대학 컴퓨터과 교수
2005년 광운대학교 컴퓨터 과학과 박사 재학 중.
<주관심분야 : 네트워크 보안, 통신, 컴퓨터>



유 황 빈(정회원)
1975년 인하대학교 전자공학과 공학사 졸업
1977년 연세대학교 대학원 공학석사 졸업
1989년 경희대학교 대학원 공학박사 졸업
1981년~현재 광운대학교 컴퓨터소프트웨어학과 교수
<주관심분야 : 멀티미디어통신 및 응용, 네트워크 보안, RFID>

한 인 성(정회원)
2001년 배재대학교 컴퓨터공학과 학사졸업
2004년 광운대학교 컴퓨터과학과 석사졸업
2004년~현재 광운대학교 컴퓨터과학과 박사재학
<주관심분야 : 네트워크 보안, RFID 보안 >