

액티브 네트워크 노드의 설계*

김동영⁰, 한민호, 이영석, 나중찬
한국전자통신연구원 정보보호기술연구본부 네트워크보안연구부
(kdy63281⁰, mhhan, yslee, njc)@etri.re.kr

Design of Active Network Node

Dong-Young Kim⁰, Min-Ho Han, Young-Seok Lee, Joong-Chan Na
Network Security Department, Information Security Technology Division, ETRI

요 약

액티브 네트워크 상의 각각의 노드는 패킷을 통해 전달된 프로그램을 수행시키는 역할을 수행한다. 본 논문에서는 액티브 네트워크 노드의 설계를 기술한다. 이를 위해 액티브 네트워크의 개념을 기술하고, 노드운영체제, 수행환경, 액티브네트워크 응용프로그램의 구조를 설명하고, 또한 응용프로그램을 개발하기 위한 프로그래밍 모델에 대하여 기술한다.

1. 서론

기존의 네트워크에서 네트워크 노드의 역할이 패킷을 목적지까지 오류 없이 전달되도록 하는 것인데 비하여, 액티브 네트워크에서는 각각의 노드가 전달되는 패킷에 대하여 계산을 수행할 수 있고, 사용자가 노드에서 수행될 프로그램을 만들 수 있다. 각각의 노드에서 프로그램이 수행되기 위해서는 각 노드에는 프로그램을 수행시켜주는 환경이 존재해야 한다.

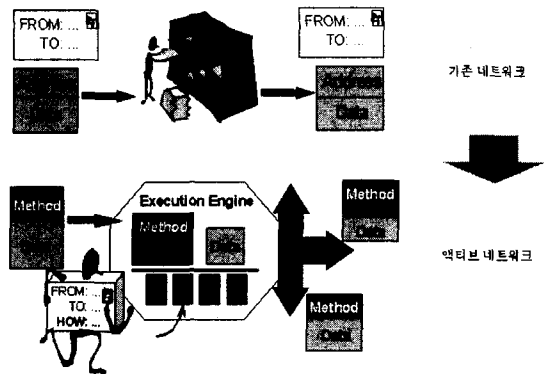
본 논문은 이러한 노드의 기능을 수행하는 액티브 네트워크 노드에 대한 설계를 기술한다.

2. 액티브 네트워크

기존의 네트워크에서 네트워크 노드의 역할은 패킷에 기록된 목적지 정보를 사용하여 그 패킷을 목적지에 도착 시키기 위해 다음의 네트워크 노드로 전송하는 역할을 수행한다.

그러나 <그림 1>와 같은 액티브 네트워크에서 네트워크 노드는 프로그램을 수행하기 위한 수행환경의 역할을 수행한다. 패킷은 패킷의 목적지와 함께 액티브 노드에서 수행되기 위한 데이터나 수행방법을 포함할 수 있다. 액티브 노드는 패킷에 대하여 필요한 동작을 수행하고 그 결과에 따라 다르게 동작할 수 있다. 액티브 네트워크에서 동작하는 코드는 노드에 미리 설치된 코드가거나 패킷을 통해 이동하면서 실행하는 코드일 수 있다.

액티브 네트워크 기술은 네트워크 노드를 융통성 있게 사용할 수 있게 해주며, 네트워크 자체를 프로그램 수행을 위한 환경인 것처럼 사용한다.

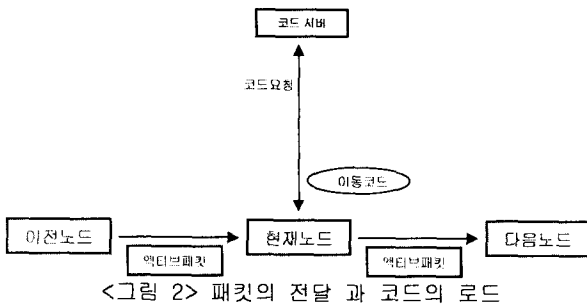


<그림 1> 액티브 네트워크

3. 이동코드의 실행 모델

액티브 네트워크 노드는 네트워크를 통하여 동적으로 코드를 이동시키고 수행할 수 있는 환경을 제공한다. 이때 이동된 코드는 자바 언어로 프로그래밍 된 코이어야 한다. 노드로 적재되어 실행되는 자바 이동코드 들은 코드서버에 의해 통합관리 되어지며, 액티브 패킷 내에는 코드서버로부터 이동코드를 적재하기위한 스크립트 언어와 수행환경과 이동코드에 의해 사용되는 데이터를 담고있다. 액티브 패킷이 노드에 도착하면 노드의 수행환경은 패킷 내에 포함된 스크립트를 해석하여 필요한 코드를 코드서버로부터 적재하여 실행시킨다. 액티브 패킷의 전달 및 수행과정은 <그림2> 과 같다.

* 본 논문에서 제안된 연구는 정보통신부연구과제로 지원되었다.



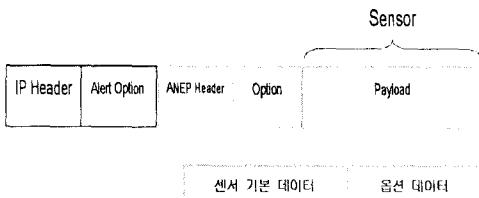
<그림 2> 패킷의 전달 과 코드의 로드

또한 수행환경에서 실행될 코드가 액티브 패킷에 의해 동작하지 않고 수행환경이 실행을 시작할 때 적재되어 실행될 수 있다.

스크립트언어는 이동코드의 적재 및 실행을 제어하기 위한 것으로 수행환경에 의해 해석되어 실행된다.

이동코드와 데이터는 <그림 3>와 같은 액티브 패킷을 사용하여 전달한다.

IP Header, Alert Option, ANEP(Active Network Encapsulation Protocol) Header 부분은 NodeOS에서 처리되어지고, Payload 부분만 수행환경으로 전달된다. 여기서 Payload 정보부분을 센서라고 한다.

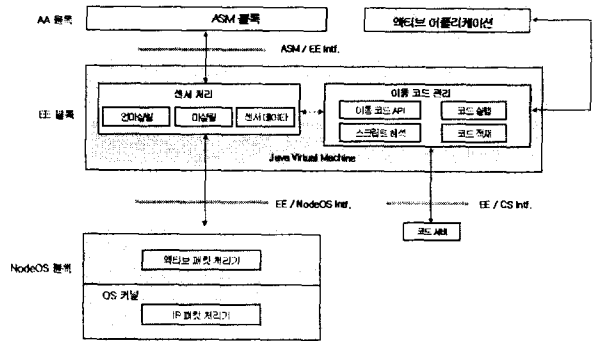


<그림 3> 액티브 패킷 구조

센서는 다시 센서 기본 데이터와 옵션 데이터 영역으로 나누어진다. 센서 기본 데이터 영역은 센서 아이디, 이전 센서 아이디, 센서타입, 생성일시, 소스주소, 목적지 주소, 자바 이동코드를 적재하기 위한 스크립트 언어를 포함한다. 옵션 데이터 영역은 센서의 종류에 따라 다른 정보가 들어간다. 센서가 다시 다른 노드로 전송될 때에는 NodeOS에서 IP Header와 ANEP Header를 생성하여 전송한다. 수행환경에서는 센서 안의 데이터들이 자바 클래스의 객체로 만들어져 이동코드에 의해 사용되어진다.

4. 액티브 네트워크 노드의 구조

액티브 네트워크 노드는 <그림 4>과 같이 노드운영체제, 수행환경, 액티브네트워크 응용의 세가지 부분으로 나누어 진다.



<그림 4> 액티브 네트워크 노드의 구조

가. 노드운영체제

노드운영체제 부분은 기본적으로 2 계층 구조를 가지고 있다. 하위계층의 IP 패킷 처리기는 OS 커널에 존재하며, 네트워크를 통해 전송된 패킷들로부터 액티브 패킷을 구분하여 상위 계층의 액티브 패킷 처리기로 전달한다. 또한 상위 계층의 액티브 패킷 처리기로부터 전달된 액티브 패킷은 네트워크를 통해 다른 노드로 전송하기 위해 IP Router Alert Option 을 호환한 IP 헤더를 생성하는 기능을 수행한다.

상위 계층의 액티브 패킷 처리기는 IP 패킷 처리기로부터 액티브 패킷을 전달 받아서 패킷 내에 존재하는 페이로드를 수행환경으로 전달한다. 또한 수행환경으로 부터 다른 노드로 전송할 데이터를 전달받아 액티브 패킷을 생성하여 IP 패킷 처리기로 전달하는 기능을 수행한다.

나. 수행환경

수행환경 블록은 센서 처리(Sensor Processor) 부분과 이동 코드 관리(Mobile Code Manager) 부분으로 구성되어 있다. 수행환경 내의 두 구성요소는 자바가상기계(JVM: Java Virtual Machine) 상에서 수행된다.

수행환경내의 센서 처리 부분은 액티브 패킷의 내용을 읽는 언마살링, 전달 가능한 패킷의 형태로 만드는 마살링, 센서데이터 관리기능을 가진다. 이동 코드 관리기는 코드 서버와 외부 인터페이스를 가지고 스크립트 언어를 해석, 이동코드를 로드하는 코드적재, 적재된 이동코드를 실행하는 코드실행, 이동코드에 의해 호출되는 API를 지원하는 기능을 가진다.

수행환경은 NodeOS, 코드서버와 인터페이스를 가진다. 코드서버는 수행환경에서 수행될 자바 이동코드를 저장관리 한다. 코드서버는 LDAP 디렉토리 시스템을 사용하여 구성되어 있다.

다. 액티브네트워크 응용

액티브 네트워크 응용은 수행환경 위에서 수행되는 응용 프로그램으로 자바 언어를 사용하여 프로그래밍된다. 설계된 수행환경에서 실행되려면 특정한 프로그래밍 모델을 통하여 설계되어야 하며 그

모델은 다음 장에서 기술한다.

액티브 네트워크 응용은 코드서버에서 저장되다가 필요한 노드로 적재되어 실행되어진다. 한번 적재되어 실행된 응용은 해당 노드에 보관되어 동일한 응용이 반복 수행될 때에는 적재 과정을 거치지 않고 바로 수행되어진다.

5. 이동코드의 프로그래밍 모델

본 시스템에서 사용하는 이동코드는 자바언어를 사용하여 작성할 수 있다. 그러나 특정한 모델에 따라 프로그램을 작성해야만 수행환경에서 적재하여 실행이 가능하게 된다. 수행환경에서는 이동코드에서 사용할 수 있는 API 들을 제공한다. 이 API 들은 액티브 노드에서 제공하는 기능들을 클래스의 형태로 제공한다. 제공되는 클래스들 중 중요한 클래스는 아래와 같이 Mcode(모든 이동코드에 대한 최상위 클래스), Sensor (액티브 패킷에 의해 전달되는 데이터를 표현), NodeManager(액티브 노드에 대한 정보를 제공)이다.

- 이동코드(Mcode)

모든 이동코드가 계승해야 하는 추상 클래스이다. 이것은 자바 애플릿을 만들기 위해서는 Applet 클래스를 계승해야 하는 것과 같다. Mcode는 이동코드를 위한 기본적인 기능, 메소드, 보안모델을 가진다. 이동코드는 McodeMain 메소드를 재정의하여 구현하여야 하며, 코드가 시스템으로 적재되면 McodeMain 메소드가 새로운 스레드로서 동작하게 된다. 수행환경에서 새로운 액티브 패킷을 받게 되면 패킷 내의 정보에 따라 이동코드를 수신하여 로드하고, 적재된 이동코드에 데이터를 전달하고 실행시킨다.

- 액티브 패킷(Sensor)

Sensor 클래스는 액티브 패킷을 표현한 것으로 액티브 패킷 안에 있는 데이터를 나타내는 클래스이다. 수행환경은 패킷 내의 데이터를 Sensor 클래스의 객체로 만들어 이동코드로 전달한다. Sensor 클래스에서는 패킷의 데이터를 읽고, 변경하고, 다시 패킷의 형태로 다른 노드로 전송하는 메소드들을 지원한다.

- 노드(NodeManager)

노드 자체를 추상화해주는 클래스로, 노드에 대한 정보를 얻고 설정할 수 있도록 해준다.

다음의 프로그램은 기술한 프로그래밍 모델을 사용하여 액티브 패킷을 목적지 까지 전달하는 간단한 예이다.

```
import eSMART.ee.*;
import eSMART.common.*;

public class Tracing extends Mcode
{
```

```
SensorInfo sin;
```

```
public void McodeMain(String[] args)
{
    sin = m_rcvSensor.getSensorInfo();

    if (NodeManager.getMyAddress().equals(
        sin.getDIA()))
    {
        System.out.println("Complete");
    }
    else
    {
        m_rcvSensor.send();
    }
}
}
```

수행환경 제어 스크립트 언어는 자바 이동코드의 적재 및 실행을 위한 정보를 전달한다. 코드의 의미는 코드서버에서 코드타입 200에 해당하는 코드의 Version 2.1 코드를 코드서버(129.254.242.44)로 부터 적재하여 실행시키는 것이다. 이때 실행되는 이동코드의 아규먼트로 "10" 이 전달된다.

```
run 200 ("10")
    after install version "2.1"
    loc "129.254.242.44";
```

6. 결론

본 논문에서는 액티브 네트워크를 위한 노드에 대한 설계와 노드에서 수행되는 응용 소프트웨어를 프로그래밍 하기위한 프로그래밍 모델에 대하여 기술하였다.

설계된 시스템에 추가되어야 할 부분은 허락되지 않은 이동코드의 실행을 막기 위해서 전달된 이동코드에 대하여 인증을 수행하는 보안기능의 추가가 요구된다.

참고문헌

- [1] 네트워크보안연구부, 한국전자통신연구원, "차세대 인터넷을 위한 능동보안기술 백서", May, 15, 2001
- [2] 나중찬, 손승원, 박치창, "액티브 네트워크 프로그래밍 언어 동향", ETRI 주간기술동향, 2001
- [3] David J. Wetherall, John V. Guttag and David L. Tennenhouse, "ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols", IEEE OPENARCH'98, Apr., 1998.
- [4] Stephen F. Bush and Amit B. Kulkarni, "Active Networks and Active Network Management", p.9-26, 2001