

능동 네트워크에서 노드 운영체제

장승주*, 이영석**, 나중찬**
*동의대학교 컴퓨터공학과, **ETRI 능동보안기술연구팀

Node Operating System in the Active Network

Seung-ju Jang*, Young-seok Lee**, Jung-chan Na **
* Dong-eui Univ., **ETRI

요 약

본 논문은 액티브 네트워크에서 자원 관리를 담당하는 노드 운영체제의 구조와 기능을 설계하였다. 본 논문에서 제안하는 노드 운영체제는 도메인, 채널, 쓰레드, 메모리, 파일 모듈로 이루어져 있다. 본 논문에서 노드 운영체제의 도메인, 채널, 쓰레드, 메모리, 파일 모듈의 설계 내용에 대해서 기술한다.

1. 서 론

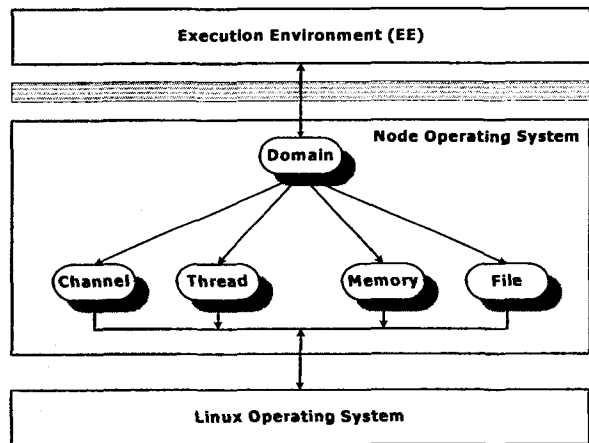
액티브 네트워크에 대한 연구는 DARPA를 중심으로 현재 여러 기관들이 자체적인 그룹을 결성하여 활발한 연구 활동을 하고 있다. 현재 주요 연구 기관 및 대학에서 연구 및 개발된 액티브 네트워크는 여러 가지가 있다. 그 중에서 ANTS는 MIT에서 연구 개발된 실행 환경과 프로그래밍 환경을 제공하고, CANE는 Georgia Tech에서 연구 중인 동적인 네트워크 변경을 처리할 수 있도록 하는 특징을 갖는다 [1, 2].

액티브 네트워크는 통일된 서비스 목표, 망 구조, 프로토콜 등의 표준이 확정되지 않은 상태에서 현재는 표준 모델 자리를 잡아가고 있는 단계이다. 액티브 네트워크의 구조는 실행 환경(EE : Execution Environment)과 노드 운영체제로 구성되며, 액티브 패킷이 실행되는 환경을 위해서 노드 운영체제 개념을 제공한다. 액티브 네트워크 사용자가 원하는 프로그램을 패킷이 가지고 있거나 중간 노드에서 유동적인 처리를 할 수 있다 [3, 4, 5].

본 논문에서 제안하는 노드 운영체제는 능동 네트워크 노드에서 효과적인 자원 관리를 위한 목적으로 설계된다. 본 논문에서 제안하는 노드 운영체제의 기능은 도메인, 채널, 쓰레드, 메모리, 파일로 구성된다 [6].

2. 노드 운영체제 구조

본 논문에서 제안하는 능동 네트워크 노드 운영체제의 구조는 크게 5부분으로 구성된다 : 도메인(domain) 관리, 채널(channel) 관리, 쓰레드(thread) 관리, 메모리(memory) 관리, 파일(file) 관리. 이 구성 요소들로 이루어진 노드 운영체제의 구조를 그림으로 나타내면 <그림 1>과 같다.



<그림 1> 노드운영체제 구조

본 논문은 2002년도 ETRI 능동보안기술연구팀의 연구 과제 지원에 의하여 이루어졌음

도메인(domain) 관리는 회계(accounting), 제어 관리(admission control), 스케줄링등을 관리하는 모듈로써 실행 환경과의 인터페이스를 담당하는 모듈이다. 각 도메인은 특수한 패킷을 실어나르는데 필요한 자원(받고 보내는 메시지, memory pool, thread pool등)을 가지고 있다. 입력 채널(input channel)에 도착한 능동 패킷은 도메인에 할당된 메모리와 쓰레드를 사용하여 실행 환경에 의하여 처리된다. 그리고 이것은 출력 채널(output channel)로 전송된다. 도메인은 계층적으로 구성이 가능하다.

도메인은 패킷을 주고 받기 위해서 채널을 생성한다. 어떤 채널은 실행 환경에서 키 역할을 수행한다. 이 채널은 EE와 하부 구조 사이에 패킷을 전달하는 역할을 수행한다. 이 채널은 입력 채널과 출력 채널 역할을 수행한다. 채널은 능동 네트워크에서 중심 역할을 수행한다. 컷 채널은 EE에 의하여 처리되거나 방해되어지지 않는 패킷으로 입력 디바이스에서 출력 디바이스로 전송이 된다.

능동 네트워크에서 계산을 위한 기초 모듈로 쓰레드 개념을 사용한다. 각 도메인은 도메인이 생성될 때 하나의 쓰레드 풀(pool)을 갖는다. 쓰레드 풀은 도메인이 생성될 때 초기화된다.

능동 네트워크에서 메모리 모듈은 패킷 버퍼의 수행에 사용되어 지고 실행 환경의 상태를 저장하는데 이용된다. 메모리 풀은 콜백(callback) 함수를 가지고 있다. 이 콜백 함수는 자원의 한계를 초과하게 되면 노드 운영체제는 이 함수를 호출하게 된다. 또한 노드 운영체제가 자원의 한계를 초과하게 될 경우 실행 환경에서 현재 상황을 처리할 수 있도록 요구한다.

파일 모듈은 POSIX 1003.1 인터페이스와 계층적인 구조를 지원한다. 제안한 노드 운영체제의 하부 운영체제는 Linux를 사용한다.

본 논문에서 제안하는 노드 운영체제는 리눅스 운영체제 위에서 동작을 한다. 기존의 리눅스 운영체제의 기능을 이용하여 능동 네트워크의 기능을 제공하기 위한 노드 운영체제로써의 역할을 수행한다. 노드 운영체제는 기존의 리눅스 운영체제 기능을 활용하여 능동 네트워크 환경에서 동작이 가능한 모듈로 설계한다.

3. 자원 관리 기법

노드 운영체제에서 실행 환경과의 인터페이스는 도메인

모듈이 담당하고 있다. 도메인 모듈은 노드 운영체제의 전체적인 기능을 관장하는 역할을 수행한다. 3장에서는 도메인 모듈의 자원 관리를 위한 기능을 중심으로 설명하고자 한다. 노드 운영체제의 도메인 모듈에서 자원 관리는 도메인의 생성 및 제거를 중심으로 이루어진다. 노드 운영체제에서 도메인의 생성 및 제거에 대해서 설명하면 다음과 같다. 도메인 생성은 새로운 도메인을 하나 만들고자 할 경우 실행환경에서 노드 운영체제의 도메인 생성 모듈에 도움을 요청한다. 도메인 생성 모듈의 동작 과정을 나타내면 <그림 2>와 같다.

도메인 생성 과정은 먼저 새로운 도메인 구조체 할당을 위하여 domain구조체를 생성하고, Cred 구조체를 생성한다. 도메인이 할당되지 않았을 경우는 오류를 반환한다. 정상적인 도메인 구조체를 할당받은 경우는 도메인 구조체 내에 descriptor table구조체를 생성한다. 할당된 도메인 구조체내에 descriptor table구조체 할당이 정상적으로 이루어지지 않은 경우는 오류 처리 후 반환한다. 정상적인 descriptor table구조체가 할당된 경우 오류 점검을 통해서 오류가 없을 경우는 현재 할당된 도메인 구조체 domain을 extern_domain으로 할당한다.

도메인 제거 모듈의 동작은 먼저 인터럽트를 실행중지(disable) 시키고, 현재 도메인의 구조체 domain을 구한다. 이 domain 구조체에서 descriptor table포인터를 구한 후, 도메인을 해제(release)한다. 그런 다음 descriptor table을 반환하고, 인터럽트를 실행가능(enable) 하게 한다.

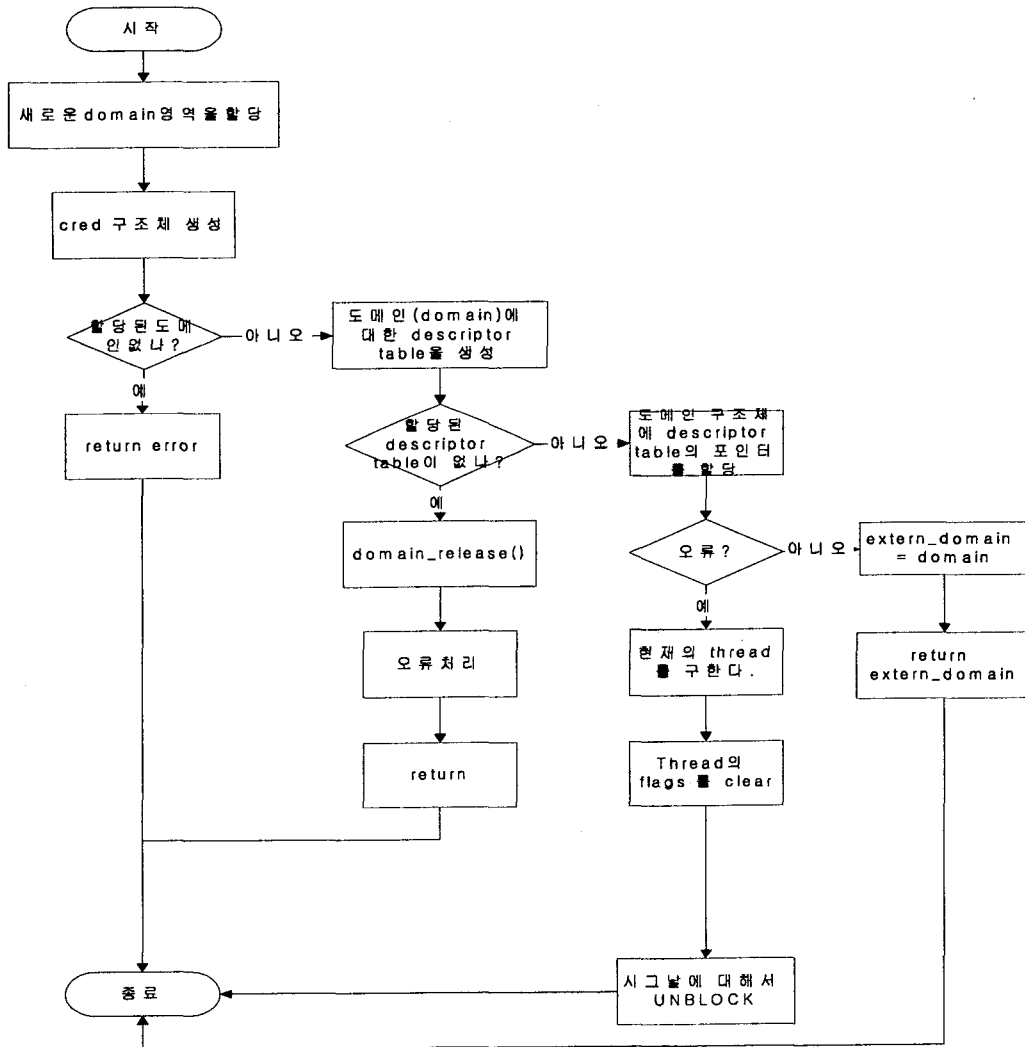
4. 결 론

본 논문은 액티브 네트워크에서 자원 관리를 위한 핵심적인 부분인 노드 운영체제의 설계 내용에 대해서 설명하였다. 본 논문에서 제안하는 노드 운영체제는 도메인, 채널, 쓰레드, 메모리, 파일 모듈을 중심으로 기술하였다. 특히, 실행 환경과의 중요한 인터페이스 역할을 하는 도메인 모듈은 노드 운영체제에서 중요한 역할을 수행한다. 그리고 실제 실행 환경에서 요구하는 자원 관리에 대해서 패킷 실행을 담당하는 채널 모듈은 내부적인 동작 과정에서 핵심적인 역할을 수행하는 부분이다. 나머지 쓰레드, 메모리, 파일 모듈은 도메인과 채널이 동작하는데 필요한 보조 역할을 수행한다.

참고 문헌

[1] 이중수, 이승현, 이영희, “Active Network 구조 : 문제점 및 접근 방법”, Sigcomm Review, pp. 109-126, 2000.12.
 [2] 이남희, “Active 네트워크 기술 개발 현황”, Sigcomm Review, pp. 127-142, 2000.12.
 [3] Calvert, K. et al, “Architectural Framework for Active Networks”, an Working Group Draft, July

1999.
 [4] L. Perterson. Et al, “NodeOS Interface Specification”, an Node OS Working Group, Feb. 1999.
 [5] Danny Raz and Yuval Shavitt, “Active Networks for Efficient Distributed Network Management”, IEEE Communications Magazine, March 2000.
 [6] 한국전자통신연구원, 차세대 인터넷을 위한 능동보안 기술 백서, 한국전자통신연구원, 2001.5.15.



<그림 2> 도메인 생성 과정