

# 액티브 네트워크 노드의 다중 수행환경을 제공하기 위한 기법 연구

김동인<sup>○</sup> 이병기 조국현  
광운대학교 컴퓨터과학과

## Study on method to provide multiple Execution Environment to nodes over Active Network

Dongin Kim<sup>○</sup> Byungki Lee Kukhyun Cho  
Dept. of Computer Science, Kwangwoon Univ.

### 요 약

액티브 네트워크는 사용자의 급변하는 다양한 요구에 대응할 수 있는 서비스중심의 네트워크이다. 기존의 네트워크가 단순한 패킷 전달의 역할만을 담당했다면, 액티브 네트워크는 사용자가, 제공되는 망의 서비스 외에 필요한 다른 서비스를 구성, 설치함으로써 학문적 또는 상업적으로 좀 더 능동적으로 망을 사용할 수 있게 해준다. 이러한 액티브 네트워크에서 기존의 제공 서비스들(패킷 전달, 라우팅, 전자메일 등)과 함께, 사용자 혹은 망 관리자가 제공하는 서비스들이 공존하기 위해서는 망에 다양한 수행환경 (Execution Environment)이 존재해야 한다. 본 논문에서는 액티브 네트워크상의 노드들에 위치하는 다양한 수행환경을 구성하고, 이를 관리하는 기법과 다중 수행환경을 이용한 다중 가상 네트워크에 대해, 현재 시도되고 있는 실 예들을 분석하여 보고, 효율적인 수행환경 구성을 위한 조건을 제시한다.

### 1. 서론

현재의 인터넷이 전 세계의 모든 정보통신 발전의 주역을 담당하고 있음에는 틀림이 없지만, 망 사용자의 다양한 욕구를 충족시키기에는 아직 요원한 점이 많다. 사용자들이 원하는 서비스가 인터넷을 비롯한 각종 망의 표준이 되려면 연구, 설계, 적용등의 과정을 거치는 많은 시간이 소요된다. 설사 그만한 작업을 거쳐 표준이 된다 하더라도 작은 규모의 그룹이나 개인적으로는 불가능한 작업임에 틀림이 없다. 이렇게 정적인 망에 좀 더 능동적인 역할을 부여하는 것이 액티브 네트워크이다. 하지만 액티브 네트워크가 능동적인 역할을 한다고 해서, 기존의 모든 망이 제공하는 서비스를 무시하고 새로 시작할 수는 없다. 기존 패킷이 요구하는 다양한 서비스 외에, 액티브 네트워크에서 사용자가 구성한 서비스를 제공하기 위해 다양한 수행환경이 필요하다. 본 논문에서는 각 액티브 네트워크에 도착하는 패킷에 서비스를 제공하는 다양한 수행환경을 액티브 노드상에 구성하는 방법과 각 패킷들이 그 다양한 수행환경에 전달되는 과정을 살펴보고 분석하여, 효율적으로 이를 구성하는 방법을 알아본다.

### 2. 액티브 네트워크

#### 2.1 개요

서론에서 액티브 네트워크가 다양한 서비스를, 사용자 또는 망 관리자가 액티브 노드에 구성하여, 급변하는 다양한 요구를 충족시키기 위한 망이라고 언급하였다. 이러한 점에서 살펴볼 때 기존의 망과 가장 다른 점은 바로 액티브 네트워크가 programmable하다는 것이다. 사용자는 원하는 서비스가 있을 경우 그에 해당하는 프로그램 혹은 데이터를 전송하면, 그 서비스는 적절한 인증과정을 거친 후에 액티브 노드에 설

치되고, 다음 부터 전송되는 패킷은 이러한 서비스를 이용할 수 있게 된다. 이러한 방식으로 액티브 네트워크에서 사용되는 패킷을 capsule[1]이라고도 한다.

#### 2.2 관련 연구

현재 시험적으로 액티브 네트워크를 구성하여 이용해 볼 수 있는 몇가지 구현이 나와있는데, 그 중에 가장 대표적인 것이 펜실베니아 대학의 PLAN[3]과 MIT의 ANTS[4]이다. PLAN은 액티브 네트워크에서 제공하는 서비스를 구성하기 위한 하나의 프로그래밍 언어이고, ANTS는 사용자가 자신의 패킷을 전송하는 것을 커스터마이징할 수 있게 해주는 프로토콜 프로그래밍 모델이다. 두 가지 모두 toolkit이 존재하여 간단한 액티브 네트워크환경을 구성할 수 있도록 하고 있다. 이외에도 NetScript, SmakrPacket같은 모델이 있다.

#### 2.3 전송 프로토콜

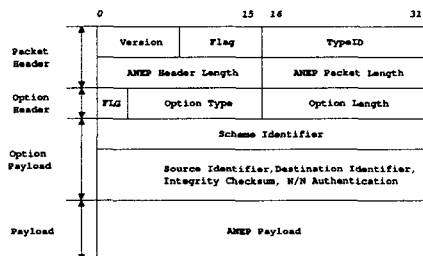


그림 1: ANEP 패킷 구조

현재 액티브 네트워크상에서 capsule을 전송하기 위해 TCP/IP와 같은 규정 프로토콜을 이용하도록 되어 있는데, Active Network Working Group에서 제안한 ANEP (Active Network Encapsulation Protocol)[2]이 그것이다. ANEP 패킷 구조는 [그림 1]과 같다. 위의 구조에서 TypeID가 여러 가지 형태의 액티브 네트워크 패킷 형태를 구분해 준다. 현재는 PLAN, ANTS의 패킷 형태만을 지원해 주고 있다.

### 3. 수행환경(Execution Environment)

#### 3.1 개요

수행환경이란 액티브 노드상에 도착하는 패킷이 처리되는 환경을 말한다.[5] 자바에서 말하는 일종의 VM이라고도 할 수 있다. 2.2절에서 언급한 PLAN이나 ANTS에서 사용하는 toolkit들은 ANEP 패킷에 encapsulation 되는 각각의 고유한 패킷 형식을 가지고 있는데, 이러한 패킷과 기존의 망에서 제공하는 TCP/IP기반의 패킷형태를 액티브 노드상에서 처리하기 위해서는 각각의 패킷을 처리할 수 있는 환경이 제공되어야 한다.

#### 3.2 구성 환경

일반적으로 사용자별로 수행환경이 download된 후에 process를 생성하는 형태를 가지므로, 대개 Unix 계열의 환경을 가지는것이 바람직하다. 현재 구현 되어있는 대부분의 액티브 네트워크 toolkit의 경우도 Unix와 linux를 기본 환경으로 하고 있다. 또한 액티브 네트워크의 성격상 프로그램 코드의 mobile성향이 강하기 때문에, 사용언어로 대부분 Java를 채택하고 있다.

#### 3.3 액티브 노드상의 다중 수행환경 구성 기법

한 개의 액티브 노드는 여러 가지의, 액티브 네트워크가 제공하는 서비스를 다룰 수 있어야 한다. [그림 2]에 다양한 패킷을 처리하는 수행환경의 구성이 나와있다.

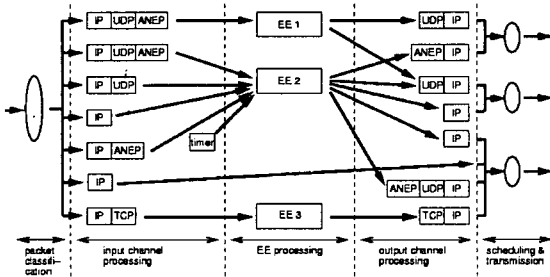


그림 2: 액티브 노드의 수행환경을 통과하는 패킷

각 수행환경을 제어하는 기법에는 2가지가 있다. 첫째는 노드 OS[5]차원에서 input, output channel을 통해서 packet flow를 각 수행환경에 할당하는 것이고, 둘째는 기존의 노드 구조에 수행환경을 제어하는 daemon만을 추가하여, 패킷들을 할당하는 방법이다. 전자는 기존의 노드의 구조를 전부 새로 바꾸어야 하는 단점이 있는 반면, 액티브 네트워크가 programmable하기 때문에 발생할 수 있는 security문제를 노드 OS 차원에서부터 해결할 수 있다는 장점이 있다. 후

자의 경우엔 비교적 큰 노력을 들이지 않고도 다중 수행환경을 구성할 수 있는 장점이 있다.

#### 3.4 수행환경의 관리

앞서 언급했던 2가지의 수행환경 구성방법중 노드 OS구성에 비해 비교적 구성하기 수월한, daemon을 이용하여 packet을 수행환경에 mapping하는 방향으로 살펴보자.

##### 3.4.1 구성(downloading and installing EE)

사용자가 원하는 수 많은 수행환경들이 각 액티브 노드상에 존재하지 않으면 도착하는 패킷에 대한 서비스 처리가 불가능하다. 하지만 수 많은 서비스들을 모두 가지고 있을 수 없으므로, 존재하지 않는 수행환경은 특정 위치에서 가져와야만 한다. 이러한 방식의 대표적인 예로 SRI의 CSL에서 제공하는 수행환경 mapping daemon인 'Anetd'[6]의 동작을 살펴보면 [그림 3]과 같다.

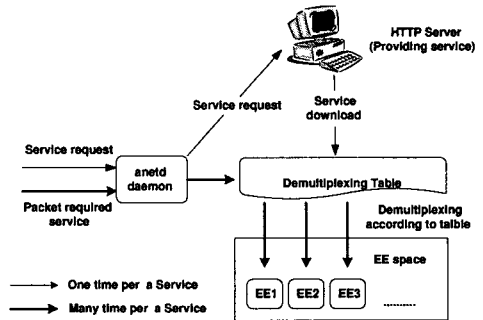


그림 3: Anetd의 수행환경 downloading

액티브 노드상에 위치한 Anetd daemon은 패킷이 특정 수행환경을 요구할 경우 지정된 서버에서 HTTP프로토콜을 통해 수행환경을 download한다. 이렇게 download된 수행환경은 Anetd가 새로운 process를 fork하면서 비로소 패킷처리의 능력을 갖추게 된다. 또한 각 수행환경은 기본 제공 서비스들을 가지고 있는데, 이러한 서비스들 역시 fork된 프로세스안에서 thread의 형태로 수행이 가능하다. 이러한 방법은 공간적인 복잡도는 줄일 수 있지만, download 할때 걸리는 시간적인 부하가 생긴다.

##### 3.4.2 해제(releasing EE)

실행이 끝난 수행환경들은 더 이상 해당 액티브 노드에 존재할 필요가 없어진다. 따라서 패킷을 전송한 application 또는 사용자가 해당 process를 제거하면 수행환경은 액티브 노드상에서 해제 된다. 하지만 여기에서 약간의 고려할 점이 생긴다. 어떠한 수행환경의 사용이 끝나서 해제한 뒤에 다른 사용자의 패킷이 동일한 수행환경을 요구하게 되면, 또 다시 download 받아야 하는 중복이 생긴다. 이를 해결하기 위해 최초 download된 수행환경의 부모 process를 생성한 뒤, 실제 사용자에게는 부모 process에서 fork시킨 자식 process를 제공하는 것이 이익이다. 또한 부모 process의 지속시간을, 나중에 요구할지도 모르는 패킷을 위해 결정하는 일종의 cache 기법도 필요하다.

##### 3.4.3 구현방향

이제까지 설명한 기법을 이용하여 다중 수행환경을 노드에 효율적으로 제공하기 위한 조건은 다음과 같다.

- 일정구역, 혹은 하나의 자치 시스템(Autonomous System) 단위로 액티브 네트워크 서비스 제공을 위한 서버를 가진다.
- 각 액티브 노드는 기존의 망이 제공하는 서비스와 대표적인 액티브 네트워크 서비스외에는 서비스 제공 서버로부터 download한다.
- 각 노드에는 여러가지 수행환경의 저장을 위한 global cache와 특정 수행환경내의 code및 data의 저장을 위한 local cache의 운용이 요구된다.
- 각 서버는 새로운 서비스가 추가될 때마다 주변의 서버들에게 서비스 update정보를 보내서, 서비스 제공의 동기화가 이루어질 수 있도록 한다.

위와 같은 조건하에서는 사용자가 직접적으로 구성한 서비스는 각 노드가 서비스 제공 서버에 질의를 해도 얻을 수 없으므로 이용이 불가능하다. 이를 위해서는 기본적으로 노드(혹은 daemon)는 각 노드에 서비스를 install할 수 있는 기능이 있어서, 일단 서비스의 임시적인 이용을 가능하게 하고, 장기간의 이용을 위해서는 적절한 인증절차를 거친후에 서버에 등록시켜야 한다.

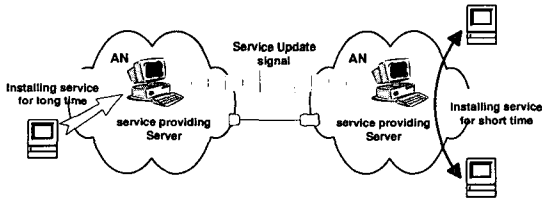


그림 4: 효율적인 다중 수행환경 구성

### 3.5 다중 가상 네트워크(Multiple Virtual Network)

사용자가 여러개의 액티브 노드에 걸쳐서 동일 수행환경을 사용하고 있다면 해당 액티브 노드를 잇는 가상 네트워크를 구성하게 되는것을 의미한다.

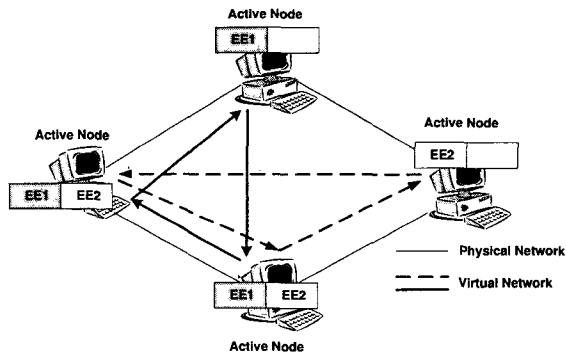


그림 5: 다중 가상 네트워크 구성

이는 MBone에서의 tunnelling의 개념과 유사하다. 실제적 으론 패킷은 여러개의 액티브 노드들을 거치지만 결과적으

로는 동일 수행환경을 지닌 노드들만이 물리적으로 이어진 것처럼 동작하는 것이다. 이러한 가상 네트워크는 특정 사용자가 액티브 노드를 독점하지 않은 상태로, 각 사용자마다의 수행환경을 제공하므로 유연성을 가진다. 마찬가지로 각 수행환경을 해제하게 되면 가상 네트워크의 구성은 소멸된다.

### 4. 결론

본 논문에서는 액티브 노드상에서의 효율적인 다중 수행환경 구성 기법에 대해 살펴보았다. 액티브 네트워크의 특성상 각 노드에는 여러가지의 서비스가 존재하게 되는데, 차별화 되는 서비스를 위해 여러 개의 수행환경이 존재한다. 각 액티브 노드상에 다중 수행환경을 구성함으로써, 각 사용자만의 가상 네트워크를 구성할 수 있고, 수행환경을 관리하는 관리자로서, 기존의 노드 환경위에 간단하게 구성할 수 있는 daemon을 지정하는 방법이 있고, 좀 더 복잡하게는 기존 노드 환경을 대체하는 새로운 노드 OS를 지정할 수 있다.

액티브 네트워크의 가장 큰 과제는 기존의 망에 익숙한 사용자들에게 보다 다양한 서비스를 제공하는 것이다. 이를 위해서는 다양한 서비스의 개발과 이를 위한 여러가지의 수행환경의 설계와 구성이 절실히 요구된다 하겠다.

### 참고 문헌

- [1] D.L.Tennenhouse,Jonathan M.Smith,“Survey of Active Network Research” , IEEE Communications Magazine, January 1997
- [2] D.Scott Alexander et al, “Active Network Encapsulation Protocol(ANEP)” , Draft, July 1997
- [3] Michael Hicks et al, “PLAN: A Packet Language for Active Networks” , Proceedings of the International Conference on Functional Programming (ICFP) '98
- [4] D.Wetherall,J.Gutttag,“ANTS:A Toolkit for Building and Dynamically Deploying Network Protocols” , IEEE OPENARCH'98, San Francisco, CA, April 1998
- [5] Active Networks Working Group,“Architectural Framework for Active Networks” , August 31,1998
- [6] Livio Ricciulli,“Anetd:Active NETWORKS Daemon” , SRI International Technical Report, August 10,1998. Available at <http://www.csl.sri.com/ancors/abone/>