

# 능동 네트워크용 노드를 기술하기 위한 프로그래밍 언어 연구

양윤심<sup>°</sup>, 정준영, 최원호, 정민수  
경남대학교 컴퓨터공학과

## The study on programming language for Active Network node

Yoon-sim Yang, Jun-young Jung, Won-ho Choi, Min-soo jung  
Dept. of Computer Engineering, Kyungnam Univ.  
E-mail : [xxx@xxx.ac.kr](mailto:xxx@xxx.ac.kr), [yyy@yyy.ac.kr](mailto:yyy@yyy.ac.kr), [zzz@zzz.ac.kr](mailto:zzz@zzz.ac.kr)

### 요약

능동 네트워크는 기존의 네트워크 노드(교환기)의 역할인 데이터 교환과 전송의 기능에서 프로그램의 실행 기능이 추가된 노드를 갖춘 네트워크를 말한다 능동 노드는 프로그래밍 언어의 선택에 따라 운영환경에 많은 영향을 끼치는 부분이다. 본 논문에서는 안전한 능동 노드 기술 언어의 요구사항과 운영환경에 관해 연구하고, 이를 바탕으로 최적의 능동노드 기술용 프로그래밍 언어의 모델에 능동 네트워크 언어의 표준으로서 가장 유력한 자바 언어 모델을 제안하고자 한다.

### 1. 서론

1990년대 중반 이후 능동 네트워크란 새로운 개념의 네트워크가 소개되었고 현재 미국을 중심으로 능동 네트워크의 연구가 활발히 진행 중에 있다. 능동 네트워크란 기존 네트워크의 라우터들이 단순히 패킷의 라우팅 목적으로만 사용되었던 개념을 바꾸어 사용자가 원하는 라우터를 자신의 사용 목적에 맞게 라우터의 기능을 변경이 가능한 네트워크다. 이러한 개념의 배경에는 기존의 IP체계가 갖는 문제점에서 시작되어 전 세계적으로 퍼져있는 인터넷이 포화 상태로 되어 사용자들이 네트워크를 유용하게 사

용하기 어렵게 되었기 때문이다. 이를 극복하기 위해 사용자는 자신이 원하는 노드에 자신의 통신 특성에 맞는 형태의 노드를 구축함으로써 능동적으로 통신에 대처할 수 있도록 하여 네트워크의 성능을 높일 수 있는 방안이 요구되고 있는 것이다. 능동 네트워크에서의 주요 연구 분야는 네트워크 구조, 네트워크 노드 구조, 노드의 실행환경, 프로그래밍 언어, 그리고 실행 안전성에 관한 연구로 세분화된다. 능동 노드의 특성을 결정하는 가장 중요한 부분을 차지하는 것이 실행 환경이며, 이 실행환경의 대부분은 프로그래밍 언어의 선택에 따라 많은 부분이 좌우된다. 본 논문에서는 능동 노드 기술용 프로그래밍 언어의 연

구를 통해 최적의 능동 노드 기술 언어 모델을 제안하고자 한다. 본 논문의 이후 구성은 다음과 같다. 2장에서는 능동 노드 기술 언어의 동향 분석에 대해 살펴보고 3장에서는 능동 노드 기술 언어의 운영환경 분석과 능동 노드 기술 언어의 요구조건 분석, 마지막으로 4장에서는 능동 노드 기술 언어의 모델 구축 및 제안과 5장에서 결론과 향후 연구방향에 대한 설명으로 마무리 하도록 하겠다.

## 2. 관련연구

### 2.1 능동 네트워크의 정의

기존 네트워크는 단순히 패킷을 store-and-forward 방식으로 전달기능을 수행한다. 반면 능동 네트워크는 데이터뿐만 아니라 중간 노드에서 실행될 프로그램 코드를 패킷 내에 가지고 store-compute-forward 방식으로 전달함으로써 중간노드에서 프로그램이 실행될 수 있도록 지원한다. 네트워크 노드에 대해 프로그래밍이 가능하다는 것이 능동 네트워크가 가지는 가장 대표적인 특징이며, 이를 노드간에 전송되는 패킷은 이동코드의 특성을 갖는다. 이동코드의 기본적인 구성 요소들은 효율성, 간결성과 유연성, 그리고 저장용량이다. 이동코드란 이종의 네트워크를 통해 이동가능하며 목적지에 도착하면 자동적으로 실행되는 소프트웨어로 정의하며, 능동 네트워크 프로그래밍 언어는 이동코드를 기술하는 것이다. 이동코드가 가져야 할 기능적인 요구사항은 이동코드가 사용되어지는 목적에 따라 조금씩 차이가 있겠지만 이동성, 안전성과 보안성을 만족해야만 한다.

### 2.2 능동 네트워크의 세계동향

능동 네트워크에 대한 국내 연구는 아직 큰 진전이 없을 뿐만 아니라 활발히 연구가 추진되지 못하고 있다. 본격적으로 기술 또는 장비 개발에 나서는 연구소나 업체는 없는 것으로 파악되고 있으며 세계 동향을 파악하는 수준에 있다고 할 수 있고 일부 대학의 경우 자체적인 능동 네트워크 시스템을 제안하고 있는 것으로 발표되고 있다.

국제적으로는 미국의 MIT, 펜실베니아, 아리조나, 콜롬비아, 조지아텍 대학교와 영국의 캐임브리지 대학 등 학계를 중심으로 활발히 연구가 추진되어 오고 있으며 연구기관으로는 IEEE 와 BBN, 업체로서는 인텔, 노끼아, 루슨트 등이 중심이 되어 OpenSig 및 Open Network Architecture라는 컨퍼런스 및 워크샵을 개최하면서 관련 기술의 표준화를 추진해나가고 있다[3]. 다음은 프로그래밍 언어를 중심으로 세계적으로 대표적인 연구 동향을 요약한 것이다.

- **Switchware** : 스위치웨어는 벨코어와 펜실베니아 대학이 공동으로 연구하고 있는 프로젝트로, 액티브 패킷과 스위치렛(Switchlet), SANE(Secure Active Network Environment)이라 불리는 액티브 네트워크 기반 구조로 이루어져 있다. 프로그래밍 언어로는 자체 개발한 PLAN(Programming Language for Active Network)을 사용하였다. PLAN은 Caml을 런타임으로 사용하였고, PLAN을 컴파일하여 생성할 수 있는 SNAP라는 언어를 제공한다
- **Netscript** : NetScript는 콜롬비아 대학에서 제시한 프로그래밍 언어로 네트워크 시스템을 개발하기 위한 환경을 제공한다 동적으로 네트워크 노드에 있는 액티브 프로그램들을 배포하고 실행할 수 있고, 통신(communication) 기반의 작업을 위한 간단한 데이터 플로우 언어라는 특징이 있다.
- **ANTS** : ANTS는 MIT 대학에서 연구 개발한 것으로 액티브 노드의 실행 환경과 네트워크 프로그래밍 모델을 제공해 주는 툴킷이다. 프로그래밍 언어로는 java를, runtime environment로는 JVM을 기반으로 되어있는 통합 방식의 대표적인 예이다..
- **Smart Packets** : BBN에서 추진하고 있는 Smart Packets는 액티브 네트워크의 기술을 이용하여 망 관리 시스템의 성능과 유연성을 향상시키고자 하는 목적에서 출발한 프로젝트이다. 사용되는 언어는 C와 유사한 고급언어인 Spanner와 가상 머신의 동작을 기술하는 저급언어인 Sprocket 이다.
- **CANE** : Georgia Tech(Georgia Institute of Technology)의 CANE 프로젝트는 액티브 네트워크 상에서 특정 어플리케이션이나 새로운 서비스를 제공하기 위해 우수한 성능으로 동적인 네트워크 구성을 변경 기능을 제공하기 위한 연구에 중심을 두고

있다. 프로그램 언어로는 C 언어를 기반으로 하고 있다.

이처럼 위와 같은 프로그래밍 언어를 중심으로 액티브 네트워크의 세계적인 연구 동향을 요약한 것을 살펴보았다. 그러나 지금까지는 액티브 네트워크에 대한 기본적 개념, 필요성, 다양한 응용들을 제시하면서 점차 액티브 네트워크의 적용 범위, 주요 응용 분야, 주요 핵심 기술, 접근 방법 등에 대하여 공감대를 이루어 가고 있는 과정에 있다.

### 3. 능동 네트워크 교환기 H/W 및 S/W

#### 3.1 능동 네트워크의 언어 비교 결과

프로그래밍 언어를 설계하거나 선택하기 위한 요구사항은 유연성, 가용성, 성능, 안전성 등 네 가지이다. 이 네 가지 요구사항을 모두 만족시키지 못하는 경우가 많으므로 여러 액티브 네트워크 연구에서 새로운 언어를 설계하거나 이러한 특성을 만족하는 언어를 선택하려는 것이다. 또한 액티브 네트워크를 위한 프로그래밍 언어는 strong typing, Garbage collection, Module thinning, Dynamic loading, 플랫폼 독립적인 표현(Platform independent representation), 쓰레드(Thread), static typing 등을 제공해야 한다.

언어로써 위의 그림 [3-1]과 다음의 [표 3-1]과 같이 크게 C/C++, ML, 그리고 JAVA 이렇게 세가지로 나누어진다.

[표 3-1] 언어별 기능의 지원여부

요구조건 기반언어	Dynamic Binding	Downloadable	Platform Independence	Safety	Security
C/C++				Good	
ML	Good	Good	Good	Good	
Binary					
Java	Good	Good	Good	Good	Good

그 결과 JAVA만이 액티브 네트워크의 요구사항을 대부분을 만족시키며 탁월한 기능을 제공한다. 다음은 능동노드 언어 종에서 왜 JAVA가 선정되었는지에 대한 구체적인 특징별 설명을 하였다.

#### 가. 플랫폼에 대한 독립성

자바 코드가 생성이 되면 이것은 자바 가상 기계(Java Virtual Machine)가 수행되고 있는 어떤 환경에서도 사용될 수 있다. 이런식으로 자바코드가 생성되어서 어떤 플랫폼이든지 실행이 가능하게 되는 이식성이 강한 코드라고 말할 수 있는 것이다. 자바 프로그램이 컴파일되면 구조에 종립적인 바이트코드로 변환되어 시스템이 자바 가상기계를 수행시키는 한 어떤 시스템에서도 동작될 수 있다. 이러한 점이 자바가 각광을 받는 이유이다.

#### 나. 객체지향성과 분산성

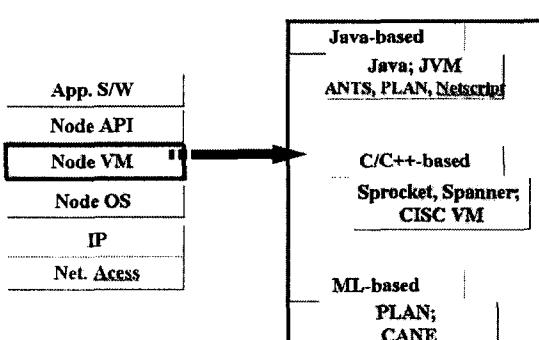
자바는 완전한 객체 지향 언어이고 TCP/IP 프로토콜을 처리하기 때문에 인터넷의 분산 환경을 위해 설계되었다.

#### 다. 동적(Dynamic) 로딩

Java 프로그램은 실행시에 객체들에 대한 액세스를 확인하고 해결하기 위해 사용된 실질적인 양의 런타임 타입(run-time type) 정보를 가지고 수행한다 따라서 안전하고 도움이 되는 방식으로 코드를 동적으로 연결하는 것이 가능하다.

#### 라. 멀티쓰레드(Multithread)

쓰레드는 이 프로세스안에 속해 있는 더 작은 프



[그림3-1] 능동 네트워크용 언어와 실행환경

능동노드 언어는 위의 연구동향에서 살펴보았듯이 액티브 네트워크의 요구사항을 대부분을 만족시키는

로세스로 쓰레드를 이용함으로써 더 효율적인 다중 작업처리가 가능하다. 자바는 대화형 네트워크 프로그램을 만들기 위한 실세계 요건과 부합되도록 설계되었다. 이 목적을 달성하기 위해 자바는 동시에 많은 일을 하는 프로그램을 작성할 수 있도록 멀티쓰레드 프로그래밍을 지원한다.

#### 마. strongly typed language

임의의 메모리 영역을 참조할 때 C와 같은 언어는 약한 타입(weakly typed) 언어이기 때문에 임의의 메모리 영역을 가리키게 되므로 부적합하다. 이는 JAVA와 같이 강한 타입(strongly typed) 언어를 사용하여 해결할 수 있다.

#### 바. 단순성

자바는 전문적인 프로그래머가 효과적으로 배우고 사용하기 쉽도록 설계되었다. 자바가 C/C++ 구문과 C++의 많은 객체지향 기능을 상속 받았고, 오히려 C++의 복잡한 더 혼동되는 개념들은 자바에서 없어지거나, 더 명백하고 더 접근하기 쉬운 방식으로 구현을 하였기 때문에 초보자들도 다른 언어들 보다 훨씬 쉽게 자바에 접근할 수 있다.

#### 사. 보안성과 이식성

자바가 보안성과 이식성 문제를 해결할 수 있는 열쇠는 바이트코드(Bytecodes)다. 자바의 런타임 시스템은 바이트코드의 인터프리터라고 할 수 있겠다. 자바 프로그램이 컴파일되지 않고 인터프리트되기 때문에 폭넓은 범위의 환경에서 훨씬 더 쉽게 실행할 수 있다. 그 이유는 자바 런타임 시스템만이 각 플랫폼에서 구현되어야 한다는 것이다. 일단 런타임 패키지가 제공된 시스템에 존재하게 되면 임의의 자바 프로그램을 그 위에서 실행시킬 수 있게 된다. 만일 자바가 컴파일 언어라면 같은 프로그램의 다른 버전들이 인터넷에 접속된 각 타입의 CPU에 따라 각각 존재해야 할 것이다. 물론 이 방법은 유연한 해결책이 아니다. 따라서 인터프리터 방식은 진정으로 이식 가능한 프로그램을 생성할 수 있는 가장 쉬운 방법이다. 자바가 인터프리트된다는 사실은 자바를 안전하게 만든다. 모든 자바 프로그램의 실행이 런타임 시스템의 제어하에 있기 때문에, 런타임 시스템은 프로그램을 포함하고 시스템의 외부에서 부작용을

일으키는 것을 방지한다. 인터프리트 된 프로그램과 컴파일된 프로그램의 실행시간은 일반적으로 인터프리트쪽이 더 느리지만 자바에서 둘의 차이는 그다지 크지 않다. 바이트코드의 사용은 자바 런타임 시스템이 기대하는 것보다 훨씬 더 빠르게 프로그램을 실행하는 것을 가능하게 한다. 또 다른 점은 자바가 인터프리터 되도록 설계 되었음에도 불구하고 기술적으로 바이트 코드를 네이티브 코드(Native Code)로 컴파일을 할 수도 있다. 그러나 비록 동적인 컴파일이 바이트코드에 적용되었다고 할지라도 런타임 시스템이 여전히 실행 환경을 책임지고 있기 때문에 이식성과 안전 기능은 적용될 것이다. 즉, 자바 이외의 다른 언어를 자바언어와 같이 실행 가능하도록 만들 수 있다는 것이다.

### 4. 능동 노드의 모델 제안

#### 4.1 PVM기반의 능동 네트워크 모델

현재 자바 가상 머신은 플랫폼의 특징과 기능에 따라서 여러 가지로 분류되어 질 수 있다. 현재는 선에서 OAK프로젝트를 통하여 최초로 개발된 JVM이 발전한 형태인 J2SE가 있고, 분산환경과 기업환경과 같은 규모 큰 API를 가지는 J2EE, 임베디드환경과 같이 소규모의 자바 실행 환경을 제공하는 J2ME로 나뉘어 진다.

능동 노드의 빠른 실행 환경을 고려할 때 데스크탑을 기준으로 개발된 J2SE의 JVM은 자바 가상 머신을 구동시키기 위해서 너무 많은 API와 부피가 큰 실행환경을 우선 메모리에 로드해야 된다는 부담과 자바의 가장 큰 취약점이라고 할 수 있는 느린 실행 환경의 제약사항 때문에 능동 노드에 탑재되는 가상 머신으로는 부적당하다고 할 수 있다. 소규모의 실행 환경을 제공하는 J2ME의 모체라고 할 수 있는 퍼스널 자바(PVM)이 적당하다고 할 수 있다. 퍼스널 자바(PVM)은 JDK1.1.8을 바탕으로 구현되었으며 대부분의 필요한 기능을 포함하고 있다. 메모리의 사용량을 최대한 줄이기 위하여 런타임 메모리를 최소화하는데 목표를 두었으며 필요한 클래스들을 미리 로드하거나 동적으로 로드함으로써 메모리의 사용량

을 줄인다. J2SE가 지원하는 대부분의 라이브러리를 지원하면서 API의 크기와 VM의 실행환경은 최대한으로 줄이는 방법을 제공한다. Net, Rmi 라이브러리를 지원하여 패킷 전달에 필요한 네트워크를 통한 통신을 가능하게 할 뿐 아니라 자바의 느린 실행을 극복하고, 직접적인 하드웨어 컨트롤을 위하여 JNI라는 라이브러리까지도 지원하고 있다.

본 논문에서 제안하는 능동 노드 기술 언어인 자바는 class라는 이동가능 코드를 생성하여 네트워크를 통해 전달될 수 있으며 능동 노드에 탑재된 자바 가상 머신에 의해서 패킷(Packet)의 일부인 실행가능 코드가 해석(interpret)되면서 강력한 기능을 제공하도록 확장될 수 있다. 또, PVM 자체에 내장되어 있는 SecurityManager를 이용한 자바보안모델은 혹시라도 있을지 모를 잘못된 실행가능코드(class)들이 능동 노드에서 실행되는 오동작을 미연에 방지할 수 있다. 다른 언어와는 구별되는 자바 언어만의 특징이라고 할 수 있는 이러한 안정성은 네트워크를 통해 전송된 class의 각종 기본 정보, 인터페이스, 필드, 메소드 정보들이 올바르게 구성 되어 있는지를 검증하는 classverifier를 PVM내부에 내장하고 있기 때문이다. 그리고, 자바언어 자체에서도 기본적인 예외사항과 예외에 대한 처리 메커니즘을 가지고 있고 속성(Attribute) 및 멤버변수(Member Var)에 대한 접근 키워드를 제공함으로서 하드웨어에 대한 무분별한 접근으로부터 보호 받을 수 있다.

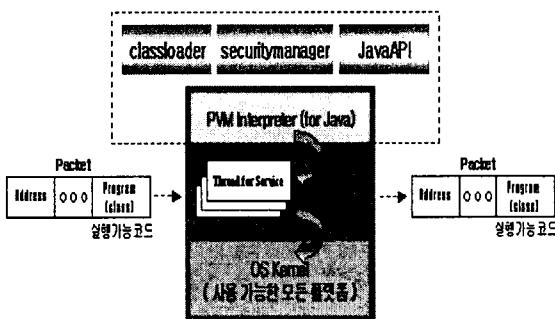
드라고 하는 능동 노드에 도달하게 되면 가상 머신의 인터프리터에 의해서 패킷의 일부인 실행가능코드들이 해석되어 진다. 해석(interpreting) 도중 필요 한 클래스들은 자신의 시스템이나 네트워크를 통해서 다운로드(downloading)되어 져야 하는데 이러한 역할을 수행하는 것이 [그림 4-1]의 'ClassLoader' 라 불리는 클래스의 defineclass, loadclass메소드에 의해서 실행된다.

자바는 자바 가상 머신과 자바 언어적인 측면에서 가장 안전한 언어라고 할 수 있다. [그림 4-1]의 SecurityManager는 조건에 맞게 자원에 접근을 허용하는데 극도로 유연하고 강력한 방식을 보장해 주고 있다. 접근을 체크하는 SecurityManager의 메소드는 조건적 접근 제어 정책을 구현하는데 필요한 인자들로 넘겨지는데, 이는 코드의 출처(로컬이나 네트워크) 결정하는 실행 스택을 체크하는 능력을 가지는 것과 마찬가지이다. 잠재적으로 위험성을 가지고 있는 실행가능 코드들에 대하여 시스템 자원에 대한 무분별한 접근을 보호하는 역할을 수행한다. 또, 자바 언어적인 측면에서 C나 C++과는 다르게 코드내부에 예외 및 에러처리를 담당하는 루틴을 작성할 수 있다.

지금까지 ActiveNetwork를 위한 이동 코드 언어들이 가져야 할 기본적인 특징을 바탕으로 플랫폼과 프로그래밍 언어측면에서 자바에 대하여 살펴보았고, 이러한 이유로 ActiveNetwork를 위한 언어로 자바가 대표적이라고 할 수 있다.

## 5. 결론

능동 네트워크가 할 기능적인 요구사항은 이동성(portability), 안전성(safety), 보안성(security)과 같은 공통된 조건들을 충실히 만족시킬 수 있어야 한다. 이에 본 논문에서는 플랫폼 독립성이 뛰어난 JAVA와 JVM을 기반으로 한 능동 네트워크의 모델을 제안하였다. 특히 PJAVA는 KVM과 비교 했을 때 우선 보다 많은 리소스를 가진 디바이스에 맞게 설계된 것이다. 따라서 KVM에서 지원하지 못하는 대부분을 지원하고 있으며 이에 따라 기존의 JDK1.1.\* 환경에서의



[그림 4-1] PVM을 이용한 능동 네트워크 제안모델

패킷들이 네트워크를 통하여 이동하면서 중간 노

작업과의 이식성, 유연성, 재사용성이 뛰어나다. 능동 네트워크는 아직도 해결해야 할 많은 기술적 문제들을 안고 있다. 성용화되기에는 아직도 많은 기간이 소요될 것으로 예상되며, 특히 기존 망과의 연동 및 backward compatibility가 상당한 부담으로 작용할 것으로 생각되며. 또한 망을 개방함으로써 나타나는 안정성 및 보안성 문제가 능동 네트워크의 성공을 좌우할 주요 요인이 될 것으로 판단된다.

### [참고문헌]

- [1] David L. Tennenhouse & David J. Wetherall, "Towards an Active Network Architecture", Computer Communication vol. 26. no. 2. 1996.
- [2] David L. Tennenhouse & David J. Wetherall, "The Active IP Option", In Proceedings of the Seventh ACM SIGOPS European Workshop, Sept. 1996
- [3] 이수형, 남택용, 나종찬, 손승원 "능동 네트워크 기술 동향"
- [4] 한국전자통신연구원 차세대 인터넷을 위한 능동 보안 기술 백서 2001.5
- [5] 한국정보처리학회 Sigcomm Review : Active Network와 Security 기술 기반 2000.12
- [6] 세계 Active Network 연구 이남희 1999.11
- [7] Active Networking 기술 소개 및 연구 현황 이종화
- [8] Survey of programmable networks 정준구 2000.3
- [9] Carnegie mellon University IEEE Communication magazine:A Programmable Router Architecture Supporting Control Plane Extensibility 지정훈 2000.3
- [10] <http://cnlab.icu.ac.kr/active.html>
- [11] <http://www.darpa.mil/ito/research/anets>
- [12] <http://www.cs.arizona.edu/liquid/> Liquid Software: A New Paradigm for Networked Systems at the University of Arizona in Tucson, Arizona
- [13] SwitchWare at University of Pennsylvania and Bell Communications Research  
<http://www.cis.upenn.edu/~switchware/>
- [14] Dynamic Integrated Resource Management (DIRM) project at BBN <http://www.dist-systems.bbn.com/projects/DIRM/>
- [15] Smart Packets at BBN : <http://www.net-tech.bbn.com/smtpkts/smtpkts-index.html>
- [16] FAIN - Future Active IP Networks at Information Society Technologies(IST) : <http://www.ist-fain.org/>
- [17] IEEE P1520 : IEEE standard for Application Programming Interfaces for Network : <http://www.ieee-pin.org/>
- [18] OPENSIG : <http://comet.ctr.columbia.edu/opensig/document/ation/>
- [19] OPENSHA : <http://www.comet.columbia.edu/openarch/>
- [20] ANGLE System : <http://cnlab.icu.ac.kr/active.html#angle>