

主題

액티브 네트워크 기술의 현황 및 전망

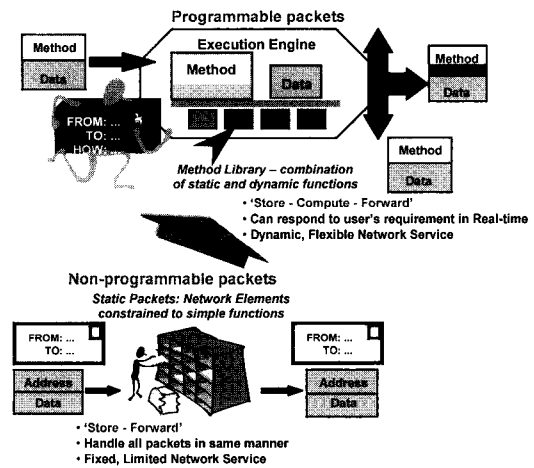
한국전자통신연구원 정보보호연구본부 오 승 희, 남 택 용, 손 승 원, 박 치 항

차 례

1. 서론
2. 액티브 네트워크 기능 구조 현황
3. 액티브 네트워크 기술을 이용한 관리 기술 현황
4. 액티브 네트워크 기술을 적용한 액티브 응용 현황
5. 액티브 네트워크 기술의 향후 전망

1. 서론

액티브 네트워크는 기존의 store-and-forward 방식의 라우터에 컴퓨팅 능력을 추가함으로써 store-compute-forward 방식으로 패킷을 처리할 수 있도록 하는 방식이다. 즉, 단순한 패킷 포워딩 기능으로만 국한된 기존 라우터에 지능화된 요소를 넣어 능력을 확장함으로써, 액티브 라우터는 기존의 라우터보다 강력한 컴퓨팅 환경을 갖추게 되며, 이와 같은 컴퓨팅 능력을 기반으로 기존 라우터에 적재되어 있지 않은 새로운 프로토콜과 기능을 동적으로 적재해서 융통성 있게 수행할 수 있게 해준다. 또한, 기존 라우터의 컴퓨팅 능력 부재로 인해 중단 시스템 기반으로만 수행되던 서비스의 경우, 액티브 라우터에서 중간 처리를 수행함으로써 기존 종단간(end-to-end) 방식에 비해 성능의 향상을 얻을 수 있게 해준다.



(그림1) 액티브 네트워크 개념

현재의 네트워크는 IP를 기반으로 모든 정보 통신 서비스를 전달하는 종합 전달망(IP over Everything)으로 발전하고 있는 상황이다. 그러나 현재의 인터넷 인프라는 새로운 응용 서비스들을 신속하게 제공하거나 새로운 기능을 추가하

기에는 적절하지 못하며, 사용자의 서비스품질 요구 수준의 기대치에 크게 미치지 못하고 있는 실정이다. 앞으로 차세대 네트워크를 구축하기 위해서는 네트워크의 성능을 저하시키지 않고 in-service 상태에서 새로운 서비스를 도입하기 용이하고, 서비스 및 네트워크의 품질 개선을 위해 네트워크의 제어와 관리를 동적으로 수행할 수 있는 새로운 네트워크 기술에 관한 접근과 연구 개발이 부각되고 있는 실정이다.

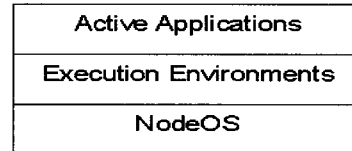
따라서 본 고에서는 현재 인터넷이 지닌 한계점을 해결하기 위한 액티브 네트워크 기술을 현재 왕성하게 이루어지고 있는 연구 활동 분야에 따라 기능 구조, 망 관리 및 보안, 응용 관점으로 분류하여 현황을 분석하고 향후 전망에 제시한다.

본 고의 2장에서는 액티브 네트워크의 기능 구조에 관련된 연구 활동을 NodeOS 측면과 EE 측면에서 살펴보고, 3장에서는 액티브 네트워크 기술을 이용한 망 관리와 보안에 관한 연구를 분석하고, 4장에서는 액티브 네트워크 기술을 적용한 액티브 응용에 대해서 멀티캐스트, 웹 캐싱, 무선, 멀티미디어로 분류하여 알아본다. 마지막으로 5장에서는 액티브 네트워크 기술의 향후 전망에 대해 제시한다.

2. 액티브 네트워크 기능 구조 현황

일반적으로 액티브 네트워크(Active Network: AN)는 Active Application(AA), Execution Environments(EE), NodeOS의 세 계층으로 이루어진다. 가장 아래층에 위치하는 NodeOS는 노드를 지나는 여러 패킷 플로우 간의 통신, 메모리, 자원 등을 제어, 관리하는 역할을 한다. NodeOS 위에는 하나 이상의 EE가 위치하는데, EE는 AA를 작성하기 위한 특정 프로그래밍 모델을 정의하며 액티브 네트워크 프로그래머들에게 기

본적인 API를 제공한다. 마지막으로 AA는 네트워크 서비스를 제공하기 위해 삽입되어진 실행 코드를 말한다.

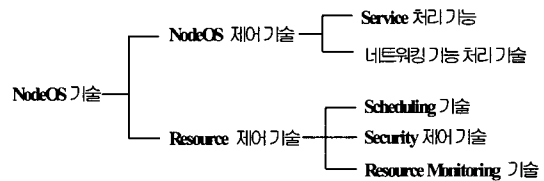


(그림 2) 액티브 네트워크 구조

2.1 액티브 네트워크 NodeOS의 개발 현황

초기 액티브 네트워크의 NodeOS는 Unix, Linux, NetBSD 등과 같은 호스트 OS를 그대로 사용하거나 이를 수정하여 사용하였다. 따라서 NodeOS와 EE간의 명확한 구분이 없었으며 per-flow 처리에 관해서도 기존 OS의 커널을 수정하여 기능을 추가한다거나 JVM 상에서 구현되었다.

액티브 네트워크에 대한 연구가 진행되면서 NodeOS와 EE간의 명확한 구분이 필요하게 되었고, 2001년 Active Network NodeOS Working Group에서는 NodeOS Interface Specification[1]을 발표하여 NodeOS에서 제공해야 하는 일반적인 인터페이스를 정의하였다. 이를 바탕으로 NodeOS에 관한 여러 연구가 진행되고 있는데 대표적 연구로는 Java OS를 NodeOS로 이용하기 위한 연구인 Janos[2]와 Joust[3], POSIX를 지원하는 유닉스 시스템을 위한 Bowman[4], NetBSD를 NodeOS로 사용하는 ANN[5], Router Plugin[6] 프로젝트, 리눅스를 사용하는 PAN[7]이 있다.



(그림 3) NodeOS의 주요 세부 기능 요소

○ Janos

Janos는 강력한 자원 관리와 Java로 쓰여진 신뢰할 수 없는 AA의 제어에 초점이 맞춰져 개발된 NodeOS이다. Janos는 low-level NodeOS, resource aware JVM, AN protocol EE의 주요 구성요소를 포함하며, 이들 각 요소는 상호 분리되어 사용 가능하다. Janos AA는 ANTS 기반의 ANTSR r실행시간 상에 작성되며, ANTSR은 JanosVM이라 불리는 JVM 상에서 동작한다. 이들 두 요소가 EE 계층을 구성하며 Moab라 불리는 Janos NodeOS 상에서 동작한다. JanosVM과 ANSTR 실행시간은 Java runtime과 매우 유사한 환경에서 AA를 지원하나, ANSTR이 보다 좁은 범위의 표준 라이브러리를 제공한다는 점이 다르다. ANTSR API는 AA 프로그래머에게 동적이며 온디맨드 코드 로딩, 비명시적 packet-matching key의 등록, 패킷의 dispatch를 위한 메커니즘 등을 제공한다.

○ Bowman

Georgia Institute of Technology에서 연구된 Bowman은 AN을 위한 확장 가능한 플랫폼으로서 System V Unix와 같은 표준 호스트 운영체제 위에 AN 기능들이 위치한다. 원래 CANEs (Composable Active Network Elements) EE를 위한 플랫폼으로 개발되었으나 다른 EE들도 구현 가능하도록 일반적인 플랫폼을 제공한다. Bowman은 AN을 지원하기 위해 Channel, a-flows, state-store에 대한 기본적인 abstraction을 제공한다. 이와 더불어 Bowman은 일반적인 OS에서의 loadable module과 비슷한 확장 메커니즘을 갖고 있어 Bowman 노드의 추가 요소가 실행 시간에 동적으로 로드 되어질 수 있다. 그러나 일반 Unix 상에서 동작하기 때문에 자원 제어에 관한 메커니즘은 따로 제공하지 않는 단점을 가지고 있다.

○ Router Plug-in 프로젝트

Washington University의 Router Plug-in 프로젝트는 NetBSD를 수정하여 들어오는 패킷을 특정 플로우로 분류하여 per-flow 프로세싱을 가능하게 하였다. 그러나 채널과 같은 커뮤니케이션에 추상화 개념은 포함하지 않고, per-flow 처리 모듈이 커널 내에 위치하고 있어 NodeOS와 EE가 합쳐진 형태를 가지고 있다. 따라서 처리 속도에 있어서는 어느 정도 이점이 있으나 유연성 및 확장성에 있어서는 한계를 갖고 있다.

○ PAN(Practical Active Networks) 프로젝트

MIT의 PAN 프로젝트는 ANTS 프레임워크에 한 고성능 캡슐 기반의 액티브 노드를 만드는 위한 것으로, Java byte code를 수행할 수 있는 EE를 리눅스 커널 내에 위치시킴으로써 패킷 전달 성능 면에서는 많은 이점을 갖고 있으나 커널 내부에 EE가 위치함으로써 다양한 종류의 EE를 수용하는 데는 한계를 갖고 있다.

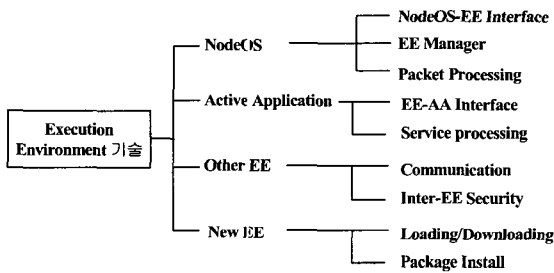
지금까지 발표된 NodeOS들도 일반적인 NodeOS의 특성을 갖췄기 보다는 일정 부분에 특화되어 만들어진 형태였다. 향후 액티브 네트워크 EE와 AA의 지속적인 연구를 위해서 NodeOS에 대한 스펙들이 명확해질 필요가 있다. 따라서 향후 NodeOS는 EE나 AA에 따라서 액티브 서비스가 노드에 저장된 형태인 액티브 노드 방식, 패킷에 직접 처리 방법을 기술하는 코드를 데이터와 함께 보내는 액티브 패킷 방식, 그리고 액티브 노드와 액티브 패킷 방식을 조합한 하이브리드 방식으로 발전할 것으로 예측된다.[8]

2.2 액티브 네트워크 EE의 개발 현황

액티브 네트워크는 다양한 연구 산출물을 연결할 수 있는 구조를 요구하고 있으며, 이를 위

해 ABone, FAIN 등의 프로젝트에서 액티브 네트워크 EE에 대한 연구가 수행되고 있다. 현재 ISI에서 네트워크 시그널링과 관리를 위한 용도로 개발한 ARP(Active Reservation Protocol)가 액티브 네트워크에 접목시키기 용이하며, 액티브 네트워크 내에서의 시그널링과 관리를 위한 용도로 적합하기 때문에 ASP(Active Signaling Protocol) EE에 ARP 프레임워크가 통합되었다. 이를 통해 AA의 능동적 실행과 설치가 가능해졌으며 좀더 세부적 기능의 제어와 관리가 가능해졌다.

(그림 4)는 EE의 주요 역할을 EE의 주요 요소에 대한 트리 형태로 정리한 것이다.



(그림 4) EE 기술의 주요 요소

현재 액티브 네트워크의 EE에 관한 연구는 구조적 프레임워크에 관련하여 많은 연구 그룹을 통해 집중되어 왔다. 그러나 대부분 개별적 환경에 대한 액티브 네트워크의 실현 가능성에 대해 한정적인 경향을 지니고 있었다. 따라서 액티브 네트워크의 가능성에 대해 구체적인 EE의 기술 요소들을 바탕으로 보다 통합적인 하나의 표준이 제시되어야 할 것이다.

특히, EE를 구조적 측면에서 NodeOS와 AA에 대한 요구 사항을 구체화하고 기능적 측면에서 NodeOS와 AA에 대한 지원 형식을 체계화해야 할 것이다. 더불어 다른 EE들 간의 통신과 보안 측면도 고려해야 할 것이다.

3. 액티브 네트워크 기술을 이용한 관리 기술 현황

본 장에서는 액티브 네트워크 기술을 이용한 네트워크 관리 기술과 액티브 네트워크 보안 기술에 대한 연구들을 분석한다.

3.1 액티브 네트워크 관리 기술

기존의 네트워크 관리용 프로토콜인 SNMP/CMIP는 관리되는 대상을 표현하는 MIB의 확장성 및 상호호환성 문제와 관리자-대리인 모델에 의한 트래픽의 집중화 문제를 안고 있다. 액티브 네트워크 관리는 실제 관리자의 개입 없이 환경에 맞게 노드의 설정을 변경하고 서비스를 추가하는 구성 관리와 결함을 복구할 수 있는 장애 관리의 자동화와 네트워크 트래픽을 조절하여 성능을 향상시키는 성능 관리와 액티브 노드 내에서의 자원에 대한 접근을 제어하는 자원에 대한 보안 관리 역할을 수행할 수 있다. 이와 관련된 연구들은 다음과 같다.

○ ABLE(Active Bell Labs Engines)[9]

ABLE은 Bell Labs에서 네트워크 관리를 위한 작업들의 효율적인 분산에 관한 연구로 진행된 프로젝트이다. 기존의 일반 IP 라우터와 액티브 패킷을 처리하는 액티브 엔진을 분리한 구조로 액티브 라우터와 그렇지 않는 IP 라우터(non-active router)가 공존하고 있는 상황에서는 관리를 위한 모듈이 액티브 노드 내에 존재한다.

○ ANCORS(Adaptable Network Control and Reporting System)[10]

ANCORS는 SRI에서 액티브 네트워크의 설계, 구성, 관리를 위한 네트워크 구조로 새로운 네트워크 서비스를 동적으로 추가하고 수정할 수 있는 방법에 초점을 두고 개발하였다. 필요한 인자

(argument)만을 가지고 소프트웨어를 인스톨하여 시스템을 변경 및 향상시키는 액티브 네트워크 특성을 기반으로 융통성 있는 네트워크 관리를 추구하고 있다.

○ AVNMP(Active Virtual Network Management Protocol)[11]

AVNMP 프로젝트는 자신의 동작을 예상할 수 있는 몇 가지 통신망 특성에 기반을 두고 General Electrics에서 시작한 프로젝트이다. AVNMP에서는 Driving Process(DP)와 Logical Process(LP)로 구성된 운영 네트워크와 예상 오버레이 네트워크로 구성되어 있다. DP는 각 소스 노드이고 가상 메시지를 생성하고 응용 프로그램을 제어하고, LP는 중간 노드 또는 목적지 노드이고 중단된 트랜잭션에 대한 복귀를 수행한다.

○ FAIN(Future Active IP Network)[12]

FAIN은 UCL이 주축이 되어 2000년부터 진행 중인 프로젝트로 액티브 노드를 기반으로 개방적이고 프로그램 가능하며 신뢰성 있는 액티브 네트워크 구조를 개발하는데 목적을 두고 있다. 이는 액티브 네트워크, 액티브 노드, 정책 기반의 네트워크 관리(policy-based network management: PBNM)와 동적인 프로토콜 지원을 통해서 액티브 네트워크 서비스 제공을 목적으로 진행되고 있다. FAIN 프로젝트는 사용자, 서비스 제공자, 응용 개발자들의 다양한 요구사항을 반영하여 설계한 것이 특징이다.

○ NESTOR(architecture for NETwork Self management and ORganization)[13]

NESTOR 프로젝트는 기존의 구성관리에서 발생하는 오류 복구 및 고비용에 대한 문제점을 해결하고자 Columbia 대학에서 개발한 프로젝트이다. NESTOR는 고도의 숙련된 운영자의 수작업

을 정책에 의해 자동화된 구성관리 작업이 가능하도록 소프트웨어로 구현한 것이다.

○ PANTS(Python Active Network Transport System)

PANTS는 기존 액티브 네트워크 구조들이 보호와 성능 향상을 위해서 정적인 언어를 사용하므로 발생하는 액티브 네트워크의 동적인 특성을 제한하는 문제를 해결하고자 시드니 대학에서 진행한 프로젝트이다. PANTS는 MIT에서 개발한 ANTS[14]를 기반으로 제안된 구조로 여러 부분이 ANTS와 유사하지만, 동적인 운용 및 관리 구조를 갖는다. 또한, PANTS는 캡슐을 통해 노드를 갱신할 수 있으므로 새로운 프로토콜을 동적으로 설치할 수 있다.

○ SENCOMM(Smart Environment for Network Control, Monitoring, and Management)[15]

SENCOMM은 액티브 네트워크 기술을 이용하여 기존의 네트워크를 관리하는 것뿐만 아니라 액티브 네트워크를 모니터링하고 관리하는 것에 초점을 맞추고 있다. 관리 모듈의 동적인 적용, 관리 응용 프로그램의 모니터링과 제어가 지원되며 네트워크 자원의 결함이나 오류에 대한 검출 및 해결을 자동적으로 지원하기 위한 목적으로 제시되었다.

앞서 언급한 액티브 네트워크를 관리 목적으로 하는 대부분의 연구들은 네트워크 관리의 5가지 기본 목표인 장애, 구성, 계정, 성능, 보안 관리들 중에서 지능적인 특성이 요구되는 구성, 장애, 성능, 보안 관리에 초점을 두고 있다.

액티브 네트워크는 기존의 네트워크가 가지고 있는 비효율성, 수동성, 보안, 성능, 신기술과 새로운 서비스 개발 등의 문제점을 개선할 수 있는 새로운 접근 방법이다. 또한 네트워크 확장으로 인하여 네트워크 관리에 대한 필요성이 증가

함에 따라서 액티브 네트워크 관리에 대한 연구가 많아지고 있다. 따라 많은 대학과 연구소에서 액티브 네트워크에 대한 연구가 진행 중이며, 기존의

(표 1) 액티브 네트워크 관리 연구 현황

	정의	관련 연구 프로젝트
구성 관리	관리자의 개입 없이 액티브 노드의 구성을 자체적으로 설정할 수 있는 것으로 Self-Configuration 또는 자동 구성 방식 등의 방향으로 연구 진행 중임	ANCORS, FAIN, NESTOR 등
장애 관리	결함이 노드 내에 탐지되면 그 결함을 분석하여 원인을 파악하고 자동으로 복구할 수 있는 것	AVNMP, FAIN, NESTOR 등
성능 관리	네트워킹 성능을 향상시키고 자원을 효율적으로 관리하는 것	AVNMP, FAIN 등
보안 관리	자원의 접근 통제를 통해 제공하는 방향으로 연구 진행 중임	ABLE, FAIN, PANTS, SENCOMM 등

그렇지만, 현재 액티브 네트워크 관리는 대부분 액티브 네트워크 기술을 이용한 기존의 망 관리에 초점이 맞추어져 있고, FAIN에서만 액티브 네트워크 자체에 대한 관리를 언급하고 있다. 그리고, 노드 내의 자동 설정 및 복구, 자원의 접근 제어 및 관리에 대한 연구는 있으나, 네트워크에서의 트래픽을 제어하고 성능을 향상시키는 부분에 대한 연구는 미흡한 실정이다. 게다가 향후 몇 년간 인터넷은 기존 네트워크와의 공존이 불가피하고, 표준화된 형태가 없기 때문에 다양한 액티브 네트워크가 함께 존재하게 된다. 그러므로 기존 네트워크 관리와의 연동 및 다른 액티브 네트워크 관리와의 연동 문제도 고려되어야 한다.

3.2 액티브 네트워크 보안 기술

액티브 네트워크는 패킷을 포워딩하는 기능만을 제공하는 기존의 수동적인 네트워크에 비해 더욱 복잡하여 위협과 공격이 훨씬 다양하고 이에 따라 보안 및 안전의 문제가 중요한 관심사가 되고 있다. 보안에 대한 중요성이 증가함에

보안 기법에 액티브 네트워크의 개념을 도입하는 액티브 보안 관리 기법에 대한 연구도 활발히 진행되고 있다.

○ PLAN(A Packet Language for Active Networks)[16]

펜실베니아 대학에 의해 수행된 PLAN은 액티브 네트워크 환경에서 액티브 패킷을 생성하기 위하여 개발된 언어로 프로그래밍 관점에서 안전성 및 보안의 문제를 다루었다. PLAN은 경량화하고 제한된 기능을 가진 프로그램을 제공하는 것을 기본 설계 방식으로 삼고 있다. PLAN 프로그램은 작기 때문에 인증이 필요하지 않지만 노드에 상주하는 서비스 루틴들은 필요하면 인증을 제공하고, 안전성 및 보안, 성능, 융통성의 문제를 다루고 있다.

○ Safetynet[17]

Safetynet은 Sussex 대학에서 수행된 액티브 네트워크를 위한 프로그래밍 언어 설계 프로젝트로, 상속성과 다수개의 서브타입, 메모리 관리를

위한 실행시간을 지원하고 자바와 유사한 객체 지향성 클래스 기반의 언어를 개발하였다. Safetynet의 목적은 액티브 네트워크에 다양한 기능을 제공하기 위한 분산 어플리케이션의 구현을 가능하도록 위한 것이다.

○ SANE(Secure Active Network Environment)[18]

펜실베니아 대학에 의해 수행된 SANE은 안전한 액티브 네트워크 구현을 위해 SwitchWare 환경에서 구현되었으며, 계층화된 보안 구조로 인증, 무결성, 암호 등의 보안 서비스를 제공한다. SANE은 무결성과 신뢰성을 주요 개념으로 하며, 정적인 무결성 검사를 하는 노드 레벨과 동적인 무결성 검사를 하는 네트워크 레벨의 구조로 이루어져 있다. 액티브 네트워크 노드의 하위 계층을 신뢰할 수 있도록 구성하고 상위 계층은 하위 계층의 무결성에 의해 보장되도록 함으로써 신뢰할 수 있는 계층 구조를 구성한다. SANE은 공개키 기반 구조, 키 확립 프로토콜, 패킷 인증, 패킷 암호화, 관리 영역, 이름체계 등의 보안 서비스를 제공한다.

○ Seraphim[19]

일리노이 대학에서 연구한 Seraphim은 다양한 보안 정책과 메커니즘을 수용할 수 있는 확장 가능하고, 재구성 가능한 보안 구조로 응용 프로그램과 사용자에게 동적으로 상황에 맞는 정책을 생성 및 수행할 수 있는 능력을 제공한다. Seraphim은 노드의 자원 요청을 가로채서 정책에 따라 자원을 허락하는 참조 모니터(Reference monitor)와 참조 모니터가 어떤 종류의 EE와도 상호 작용할 수 있도록 해주는 보안 프록시(Security proxy)로 구성되어 있다.

(표 2)는 액티브 네트워크의 보안을 목적으로 하는 연구를 시스템 구조의 측면에서 보안과 프

로그래밍 언어의 측면에서 액티브 네트워크를 보호하는 연구로 분류하여 작성한 것이다. 시스템 구조 측면에서의 액티브 네트워크 보안은 안정성을 바탕으로 설계하여 보안을 제공하는 방식이다. 프로그래밍 언어 측면에서는 잘 설계된 프로그래밍 언어를 사용함으로써 실행시간 검사를 줄이는 등 많은 보안 문제들을 해결할 수 있고 성능을 향상시킬 수 있다는 것과, 제한된 기능을 가진 경량(lightweight)의 프로그래밍 언어를 통해 SANE이나 다른 구조적인 보안의 필요성을 없애는 것에 대해 초점을 맞추고 있다.

(표 2) 액티브 네트워크 보안 연구 현황

액티브 네트워크 보안	관련 연구 프로젝트
시스템 구조 측면	SANE, Seraphim 등
프로그래밍 언어 측면	PLAN, Safetynet 등

액티브 네트워크가 지닌 보안상의 문제점들을 해결할 수 있어야만 기존 망과의 연동 또는 자체 네트워크로 생존할 수 있을 것이다. 따라서, 효율적이고 신뢰할 수 있는 형태의 액티브 네트워크에 대한 연구가 지속적으로 요구된다. 더불어, 액티브 네트워크 자체의 보안성 확보에 대한 연구뿐 아니라 액티브 기술을 이용한 네트워크 보안에 대한 연구도 요구된다.

4. 액티브 네트워크 기술을 적용한 액티브 응용 현황

액티브 라우터에서 수행되는 새로운 프로토콜이나 서비스를 액티브 응용이라 하며, 액티브 응용은 OSI 3 계층이상의 임의의 프로토콜이나 서비스가 될 수 있다. 액티브 응용으로서의 실현 가치가 있는 서비스들은 액티브 라우터에서의 중

간 처리를 통해 성능의 향상을 얻을 수 있는 것들이어야 한다.

4.1 멀티캐스트

기존의 멀티캐스트 문제는 신뢰적 멀티캐스트, 성능 향상과 QoS 등이 있다. 액티브 라우터의 능동적인 기능을 이용해서 이러한 문제들을 보다 효율적으로 처리하고자 하는 접근이 다수 제시되었다.

○ 신뢰적 멀티캐스트 문제

신뢰적 멀티캐스트 문제는 손실 패킷의 재전송과 혼잡 제어 문제로 나누어 볼 수 있다.

- 손실 패킷의 재전송 문제

손실 패킷의 재전송과 관련해서는 손실 복구를 위한 오버헤드, 피드백 폭주, 노출 문제가 있으며, 액티브 라우터를 통해서 이들 문제를 보다 효율적으로 처리하고자 하는 연구들이 나오고 있다.

- 손실 패킷의 재전송과 관련된 손실 복구 오버헤드 문제
액티브 라우터에 캐시를 두고 패킷을 저장하는 방식으로 해결 가능하다. 캐시에 저장된 패킷을 재전송함으로써 송신자나 수신자가 아닌 액티브 라우터에서 지역 재전송을 하고, 따라서 대역폭 낭비를 줄일 수 있을 뿐만 아니라 손실 복구 시간도 감소시킬 수 있다. 이러한 방법을 사용한 예로는 ARM(Active Reliable Multicast)[20], AER/NCA(Scalable Fair Reliable Multicast Using Active Services - active error recovery/nominee congestion control)[21], AERM(Active Error Recovery for Reliable Multicast)[22] 등이 있다.
- 손실 패킷의 재전송과 관련된 피드백 폭주 문제
액티브 라우터가 ACK나 NACK를 aggregate하고 NACK suppression을 함으로써 해결 가능하다. 이를 위해 액티브 라우터는 ACK/NACK 관련된 정보를 가지고 있어야 한다. 이러한 기능

은 ARM이나 AER/NCA의 경우, 액티브 라우터에서 유지되는 NACK 레코드와 REPAIR 레코드에 의해 지원되며, 이들 레코드는 NACK를 요청한 링크에 대한 정보 등을 갖고 중복 NACK를 무시하고 suppression한다.

▪ 손실 패킷의 재전송과 관련된 노출 문제

액티브 라우터에서 재전송해야 할 링크에 대한 정보를 유지함으로써 재전송을 요청한 수신자에게만 제한된 재전송함으로써 해결 가능하다. 관련 연구로는 ARM, AERM, AER/NCA 등이 있으며, 제한된 재전송의 범위는 액티브 라우터의 모든 하위 멀티캐스트 수신 노드 (예: AER/NCA) 또는 재전송을 요청한 링크로만 재전송하는 방법(예: ARM, AERM)이 있을 수 있다. 후자의 경우, 액티브 라우터에 NACK 요청 링크에 대한 정보를 유지해야 하는 오버헤드가 있으나 재전송 범위를 더 많이 한정함으로써 노출 문제를 수월하게 해결할 수 있다.

- 혼잡 제어 문제

혼잡 제어는 네트워크의 트래픽 상황에 따라 송신자의 전송 속도를 조절하는 것으로, 일반적으로 송신자가 수신자로부터의 모든 ACK를 기반으로 자신의 송신 속도를 조절한다. 혼잡 제어 기능을 액티브 라우터에 구현한 예로는 AER/NCA, EAMSA(An Active Congestion Control in ATM Multicast-Extended Active Multicast Service Architecture)[23], RMANP(Performance of Active Multicast Congestion Control-Reliable Multicast Active Network Protocol)[24]가 있다. AER/NCA에서는 최악의 수신인인 nominee를 액티브 라우터가 선정해서 nominee로부터의 ACK를 기반으로 송신자가 전송 속도를 조절할 수 있게 함으로써 송신자에서의 오버헤드를 감소시킨다. RMANP에서는 액티브 라우터에서 혼잡 상황의 탐지 및 대응을 함으로써 송신자의 전송 속도를 능동적/신뢰적으로 할 수 있는 방법을 제시하고 있다. 한편

EAMSA는 액티브 라우터에서 직접 전송 속도를 조절하는 방식을 사용하여 액티브 라우터에서의 오버헤드는 증가하지만 송신자 측의 오버헤드를 감소시키고, 수신 상태가 좋은 수신자들의 경우 고품질의 서비스를 받을 수 있도록 하는 장점이 있다.

○ 성능 향상 및 QoS 문제

성능 향상 및 QoS 문제는 멀티캐스트 라우팅 시 액티브 라우터의 기능을 사용해서 QoS를 보다 효율적으로 지원함으로써 해결하며, PIM-SM과 같은 코어 기반의 멀티캐스트 라우팅 방식에서 적당한 코어를 선택하고 네트워크 상황에 따라 이동함으로써 코어를 적절하게 선택할 수 있도록 한다. 또한 효율적인 실시간 비디오 전송을 위해 멀티캐스트 비디오 분산 스킴을 사용하기도 한다.

4.2 웹 캐싱

기본적으로 웹 캐싱에는 자치적 조직성, 적응성, 고성능성, 지능성 등의 문제가 존재하고 있으며, 액티브 라우터의 능동적인 기능을 이용해서 이러한 문제점들을 보다 효율적으로 처리하고자 하는 접근이 제시되고 있다.

○ 자치적 조직성 문제

자치적 조직성 문제는 기존 웹 캐싱(SQUID)의 수동 구성 시스템이라는 문제점을 해결하기 위해서 주변 웹 캐시 사이의 통신을 수동 구성 대신에 웹 캐시가 주변 웹 캐시를 찾으면, 자신들을 다중으로 겹쳐진 멀티캐스트 그룹으로 조직하거나, 캐싱 정책 파라미터를 이용한 방법 등을 이용하여 해결한다. 이러한 방법은 네트워크 수준에서 적용한 경우와 응용 수준에서 적용한 경우로 구분된다. 네트워크 수준에서 자치적 조직성 문제를 해결한 예는 MIT의 ANTS, Georgia Tech의 CANEs 등이 있으며, 응용 수준에서 자

치적 조직성 문제를 해결한 예는 LBNL/UCLA의 AWC(Adaptive Web Caching)[25], USC의 LSAM(Large Scale Active Middleware)[26], UTS/BT의 ALAN(Application Level Active Network) 등이 있다.

○ 적응성 문제

적응성 문제는 AWC보다 고수준의 웹 캐싱 적응성을 지원하는 것으로 컴포넌트 기반 기술을 이용하여 해결한다. 이러한 방법을 사용한 예로는 직접적으로 액티브 네트워크라는 언급은 없으나 유사한 기능을 가진 워싱턴 대학의 JAWS나 TAO 등이 있다.

○ 고성능성 문제

고성능성 문제는 멀티캐스트 기법과 고성능을 지원하는 웹 서버 등을 이용하여 해결한다. 이러한 방법을 사용한 예로는 Georgia Tech의 CANEs와 워싱턴 대학의 JAWS 등이 있다.

○ 지능성 문제

지능성 문제는 부하 균등화 기술, 애니캐스팅, 캐싱 협상 프로토콜 등을 이용하여 해결한다. 이러한 방법을 사용한 예로 중복 서버가 널리 퍼져 있을 때, 적당한 서버로 연결을 분산시키는 부하 균등화 스킴 및 애니캐스팅을 이용한 서버 부하 균등화를 위한 새로운 네트워크 패러다임 그리고 지능적인 웹 캐싱을 지원하는 캐싱 협상 프로토콜 등이 있다.

4.3 액티브 무선

현재의 무선 네트워크는 신속한 핸드오버와 네트워크 토폴로지의 변화에 빠르게 적응하는 라우팅에 대한 어려움을 가지고 있다. 이러한 문제점을 액티브 기능을 이용함으로써 이동 네트워크에서 신속한 핸드오버가 가능하게 하며, ad hoc 네트워크에서 적응적인 라우팅이 가능하도

록 한다.

이동 네트워크에서 이동 노드의 신속한 핸드오프를 지원하기 위해, 액티브 라우터는 홈 에이전트와 상대 노드(corresponding node)로 가는 바인딩 갱신 메시지를 직접 처리하는 방식을 제안하며, 대표적인 연구로서 ADS(Active Delivery System)가 있다.

또한 기지국이 존재하지 않는 동적인 ad hoc 네트워크에서 적응적인 라우팅을 지원하는 연구로는 ACIP(Active Cellular IP)가 있다. ACIP에서는 이동 노드가 MST(Minimum Spanning Tree)를 사용하여 자신의 위치를 알게 하고, 이를 위해 이동 노드는 이웃 발견 기능과 MST를 구축하는 액티브 기능을 탑재하고 있다.

4.4 멀티미디어

대용량 및 다양한 종류의 멀티미디어 브로드캐스팅은 현재의 인터넷상에서 심각한 대역폭 및 컴퓨팅 자원을 요구한다. 현실적으로 대용량의 VBR(Variable Bit Rate) 멀티미디어 데이터 전송을 위한 대역폭 보장은 네트워크 상에서 거의 불가능하므로, 전송 프로세스는 실제 가용한 네트워크 대역폭에 실시간 적응이 가능하도록 구성하는 것이 필수조건이 될 것이며, 이 때 허용 기준치내의 품질을 유지하면서 멀티미디어 전송이 되어야 한다. 따라서, in-service 상태에서 라우터의 기능을 향상시켜 멀티미디어 전송 및 트랜스코딩이 보다 효율적으로 이루어지도록 지원하는 액티브 네트워크 기술이 문제점 해결의 한 방안이 될 수 있다.

일반적으로 멀티미디어 전송 및 트랜스코딩을 효율적으로 하기 위한 액티브 라우터들은 전체 네트워크의 핵심적인 부분에 위치하며, 이들 라우터는 등록된 멀티미디어 서비스 제공자에게만 접근 허용이 가능하게 한다. 이러한 액티브 라우터는 효율적이면서도 안전하게 멀티미디어 브

드캐스팅을 수행하는 적응 프로토콜을 다운로드 받아서 각 멀티미디어 스트림의 세그먼트에 기반한 적응적 품질 제어 및 트랜스코딩이 가능하게 한다.

실제 적응 기법은 MPEG/JPEG와 같은 비디오 압축 기법에 맞추어 프레임별 종속성을 고려하거나 가용 자원 상태를 고려하여 Q-factor, 해상도, 전송 속도 등을 조절하도록 함으로써 각 액티브 라우터에서 실시간으로 지역적인 결정을 내려 처리하도록 구현하였다. 이들 적응 프로토콜을 액티브 라우터에 구현하기 위해서 PLAN이나 PLAN-P와 같은 언어를 구현하거나 커널 레벨에서 C언어로 능동 네트워크 구조를 위한 서비스 모듈을 구현하였다. 또한, 비디오 트랜스코딩이나 패킷 드롭을 위해 적절한 휴리스틱 알고리즘을 사용하고 해쉬 테이블을 사용하여 프레임 타입 및 프레임 번호 등을 저장하여 두었다가 현재 가용 자원 상태를 고려한 적응 기법의 정보로 사용할 수 있다.

5. 액티브 네트워크 기술의 향후 전망

액티브 네트워크 기술은 현재의 인터넷 확장과 앞으로의 차세대 인터넷 구축 시에 in-service 상태에서 새로운 서비스의 적용을 용이하게 하고, 네트워크의 성능을 저하시킴 없이 트래픽 제어 및 관리를 동적으로 수행하게 하는 새로운 기술이다.

본 고에서는 액티브 네트워크를 구조, 관리, 응용 관점으로 분류하여 기술 현황을 분석하였다. 액티브 네트워크 기능 구조에서는 NodeOS와 EE의 연구 현황 및 향후 요구사항에 대해 살펴 보았고, 액티브 네트워크 관리 구조에서는 액티브 네트워크 기술을 적용한 기존 망의 효율적인 관리 방식과 액티브 네트워크 보안 방식에 대한

연구 내용을 분석하였다. 또한, 사용자의 요구에 부합하는 새로운 액티브 응용을 멀티캐스트, 웹 캐싱, 무선, 멀티미디어로 세분화하여 각 분야별 액티브 네트워크 기술 적용 사례에 대해 살펴보았다.

액티브 네트워크 개념이 실현될 것으로 예상되는 시점은 향후 5~10년으로 일부 기술 분석 기관에서 예측하고 있으며[27], 따라서 현재 인터넷뿐만 아니라 차세대 인터넷상에서 액티브 네트워크 개념의 실현 가능성에 대한 고려가 요구된다.

차세대 인터넷 기술의 대표적인 예로는 MPLS(Multiprotocol Label Switching)와 IPv6가 있으며, 이들 기술은 패킷 전달을 위한 네트워크 기반 구조를 제공해주는 역할을 한다. 반면에 액티브 네트워크 기술은 이러한 네트워크 기반 구조상에서 일종의 융통성 있는 가상 머신 환경을 제공하는 것이라고 할 수 있다. 따라서 차세대 인터넷은 액티브 네트워크 기술과의 연계성을 통해 서비스와 응용의 적용을 고려해야 할 것이다. 또한, 액티브 네트워크 기술은 차세대 네트워크의 제어와 관리 및 보안을 위한 새로운 기술 분야에 유용하게 활용될 것이며, 기존의 네트워크에서 풀 수 없었던 여러 가지 서비스 및 응용을 적용 시 나타나는 난제들을 해결하는 효과적인 방안을 제시할 것이다.

따라서 액티브 네트워크 기술은 향후 인터넷과 같은 거대망에서의 시험을 통해 안전성과 보안성을 확보한 후, 차세대 네트워크에 적용할 수 있을 것으로 예측된다.

참고문헌

- [1] Larry Peterson, "NodeOS Interface Specification," DARPA AN NodeOS Working Group Draft, 1999.
- [2] Janos: A Java-based Active Network Operating System, <http://www.cs.utah.edu/projects/flux/janos/summary.html>.
- [3] Joust: A Java OS in Scout, <http://www.cs.princeton.edu/nsg/joust.html>.
- [4] S. Merugu, et. al., "Bowman: A NodeOS for Active Networks," IEEE Infocom 2000.
- [5] J. Turner, et. al., "Design of a Flexible Open Platform for High Performance Active Networks," Allerton Conference, September 1999.
- [6] Dan Decasper, et. al., "Router Plugins: A Software Architecture for Next Generation Routers," SIGCOMM'98, Vancouver, CA, Sep. 1998.
- [7] Erik L. Nygren, et. al., "PAN: A High performance Active Network Node Supporting Multiple Mobile Code Systems," IEEE OpenArch '99, March 1999, pp. 78-89.
- [8] "액티브네트워크기술 분석서", ETRI 네트워크보안 구조연구팀 2002. 11.
- [9] Danny Raz, Yuval Shavitt, "Active Networks for Efficient Distributed Network Management," IEEE Communications Magazine, 38(3):138-143, Mar. 2000.
- [10] Livio Ricciulli, et. al., ANCORS: Adaptable Network Control and Reporting Systems, SRI-CSL-98-01, <http://www.sdl.sri.com/papers/341>, 1998.
- [11] Stephen F. Bush, Amit B. Kulkarni, "Active Networks and Active Network Management - A Proactive Management Framework," Kluwer Academic/Plenum Publishers, 2001.
- [12] Celestin Brou, "Future Active IP Network," WP4-GMD-011-D3-pub, May 2001.
- [13] Amila Fernando, et. al., "A New Dynamic Architecture for an Active Network," IEEE OPENARCH, Mar. 2000.
- [14] D. Wetherall, et. al., "ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols," IEEE OPENARCH, Apr. 1998.
- [15] A. Jackson, et. al., "SENCOMM Architecture," Technology Document of BBN Technologies, Apr. 2000.

- [16] M. Hicks, "PLAN System Security," Jul 1998.
http://www.cis.upenn.edu/~switchware/papers/plan_security.ps
- [17] A. Jeffrey, I. Wakeman, "A Survey on Semantic Technique for Active Networks,"
<http://www.cogs.susx.ac.uk/projects/safetynet>
- [18] H.J. Kim, B.H. Chang, T.M.Chung, "Active Network Management Technology and Research Trend," Sigcomm Review, vol.1, no.1, Dec. 2000.
- [19] R. H. Campbell et. al., "Seraphim: An Active Security Architecture for Active Networks." IEEE OPENARCH 2000, Tel-Aviv, Israel, 2000.
- [20] Lehman L., et. al., "Active Reliable Multicast," IEEE Infocom, 1998.
- [21] Kasera, S.K., et. al., "Scalable Fair Reliable Multicast using Active Services." IEEE Network, 2000.
- [22] Baochun Bai, et al., "Active Error Recovery for Reliable Multicast," IEEE Computer Communications and Networks, 2001.
- [23] Hung Keng Pung, Bajrach, N., "An Active Congestion Control in ATM Multicast," IEEE Proceedings of ATM Workshop, 1999.
- [24] Sedano, M, et. al., "Performance of Active Multicast Congestion Control," International Workshop on Active Networks, 2000.
- [25] Floyd, S. Boulder, "Adaptive Web Caching," Cache Workshop, <http://www-nrg.ee.lbl.gov/floyd/web.html>, 1997.
- [26] J. Touch, "The LSAM Proxy Cache - a Multicast Distributed Virtual Cache," 3rd International WWW Caching Workshop, 1998.
- [27] J. Murphy, "Are you ready for active networking?," <http://tech.update.zdnet.com>, 2002.



오 승 희(Seung-Hee Oh)

e-mail: seunghee5@etri.re.kr
 1999년 전북대학교 컴퓨터과학과 이학사
 2001년 이화여자대학교 컴퓨터학과 공학석사
 2001년~현재 한국전자통신연구원 정보보호연구본부 네트워크보안구조연구팀 연구원

<주관심분야> 정보보호, 차세대 네트워크보안, 액티브 네트워크 등



남 택 용(Taek Yong Nam)

e-mail: tynam@etri.re.kr
 1987년 충남대학교 계산통계학과 이학사
 1990년 충남대학교 대학원 계산통계학과 이학석사
 1987년~현재 한국전자통신연구원 정보보호연구본부 네트워크보안구조연구팀 팀장

<주관심분야> 네트워크보안, 액티브 네트워크, 차세대 네트워크구조 등



손 승 원(Sung Won Sohn)

e-mail: swsohn@etri.re.kr
 1984년 경북대학교 전자공학과 공학사
 1994년 연세대학교 전자공학과 공학석사
 1999년 충북대학교 컴퓨터공학과 공학박사

1983년~1986년 삼성전자 연구원
 1986년~1991년 LG 전자(주) 중앙연구소 HI8mm 컴퓨터 팀장
 1991년~현재 한국전자통신연구원 정보보호연구본부 네트워크보안연구부 부장

<주관심분야> 네트워크보안, 차세대인터넷, 액티브 인터넷 등



박 치 향(Chee Hang Park)

e-mail: chpark@etri.re.kr

1974년 서울대학교 응용물리학과
이학사

1980년 한국과학기술원 전자계산학
과 이학석사

1987년 파리6대학 전자계산학과 공학박사

1974년~1978년 한국과학기술연구소 연구원

1978년~현재 한국전자통신연구원 정보보호연구본부
본부장

<주관심분야> 네트워크보안, 액티브 네트워크, 멀티미
디어시스템, 미들웨어, 모바일 에이전트 구조 등