

액티브 네트워크 관리 구조 디자인

이 정석*, 이 성운*, 유 기영*

* 경북대학교 컴퓨터공학과

e-mail : kit92@purple.knu.ac.kr

Design of Active Network Infrastructure

Jung-Seuk Lee*, Sung-Woon Lee*, Kee-Young Yoo*

*Dept. of Computer Engineering, Kyung-Pook National University

요 약

기존 네트워크상에서의 라우터나 스위치와 같은 중간 노드들에서 단순히 패킷의 헤더만을 처리하는 수동적인 기능을 개선하여, 사용자가 패킷 내에 프로그램 코드와 데이터를 함께 전송하고 중간노드에서는 필요한 프로세스를 처리할 수 있는 차세대 네트워크를 액티브 네트워크라 한다. 지금까지의 액티브 네트워크 기술 현황을 볼 때 네트워크 관리는 표준화된 관리 구조를 가지고 있지 않으며, 대부분의 연구들이 액티브 노드에 중점을 두고 있으며, 네트워크 내의 트래픽 제어나 성능 향상을 위한 네트워크 관리 구조에 대한 연구는 아직 미흡한 상태이다. 본 논문에서는 기존의 수동적 네트워크와의 연동 및 서로 다른 액티브 네트워크와의 상호 운영을 고려한 액티브 네트워크 관리 시스템의 Infrastructure 를 디자인 하였다. 본 논문에서 디자인한 액티브 네트워크 관리 시스템의 모델을 기초로 하여 보다 효율적이고, 유지 보수 및 다른 네트워크와의 통합 관리가 용이한 시스템 구성을 구현할 수 있을 것으로 기대한다.

1. 서론

최근의 컴퓨터 네트워크의 발전으로 인터넷 서비스, 전자상거래 등 응용분야가 급속히 확산되고 있다. 하지만, 네트워크의 기능적인 측면에서 그 발전 속도가 느리며, 하나의 프로토콜이 표준화 과정을 거쳐서 사용되는 기간과 비용 역시 매우 많이 들고 있다. 이러한 네트워크의 대안으로 액티브 네트워크가 제시되고 있다. 액티브 네트워크(Active Network)란 기존의 네트워크의 중간 노드인 스위치나 라우터에서와 같이 패킷의 헤더만을 보고 전송하는 단순 처리가 아니라, 사용자가 원하는 프로그램을 패킷 내에 가지고 있거나 혹은 중간 노드에 일부 프로그램을 내장하였다가 필요한 작업을 수행하도록 하는 관리 구조이다. 기존의 네트워크 상의 중간 노드를 액티브 네트워크 상에서는 액티브 노드(Active Node)라 한다. 지금까지의 액티브 네트워크 기술 현황을 볼 때 네트워크 관리는 표준화된 관리 구조를 가지고 있지 않으며, 대부분의 연구들이 액티브 노드의 실행환경(EE, Execution Environment)에 중점을 두고 있다[1][6][9]. 본 논문에서는 지금까지의 액티브 네트워크 기술을 알아보고, 효율적인 액티브 네트워크의 관리를 위한 Active

Network Infrastructure 를 디자인하였다. 2 장에서는 액티브 네트워크 관련 구조에 대해 간략히 요약하였고, 3 장에서는 액티브 네트워크 관리구조를 도메인 구성 환경에 따른 각각의 디자인 모델들을 제시하였으며, 4 장에서 결론을 맺었다.

2. 액티브 네트워크 구조(Active Network Architecture)

액티브 네트워크의 기본 개념은 네트워크를 구성하는 스위치나 라우터가 기존의 단순 패킷 전달 기능 외에 각 노드는 패킷 혹은 프로그램을 저장할 수 있는 저장 장치, 프로세스 방식, 새로운 서비스 추가 또는 삭제에 관련된 기술이 제공되는 컴퓨팅 환경을 갖추어 네트워크의 기능을 수동적인 방식에서 능동적인 처리방식으로 발전 시킨 것이다. 액티브 네트워크를 구성하는 요소들은 기존의 네트워크 패킷에 해당하는 캡슐, 이러한 캡슐을 네트워크 중간에서 처리할 수 있는 액티브 노드, 캡슐들을 생성할 수 있는 액티브 클라이언트/서버로 이루어 진다. 액티브 네트워크의 구조는 크게 실행 환경과 노드 운영체제로 구성된다.

실행 환경은 기존의 패킷 구조와 달리 프로그램과 데이터를 포함하고 있는 캡슐을 처리하기 때문에 프

로그와 데이터를 실행하고 처리하는 환경을 제공한다. 실행 환경에서는 액티브 네트워크 사용자들이 활용할 수 있는 가상머신과 프로그래밍 인터페이스를 정의한다. 사용자는 패킷 내에 프로그래밍된 명령들을 실행 환경에 전송함으로써 가상머신을 제어한다. 이러한 명령들의 실행은 일반적으로 일반 환경과 액티브 노드의 상태를 변경할 수 있으며, 일정 시간 후에 실행 환경이 패킷을 다른 노드로 전송할 수 있도록 한다.

노드 운영체제는 실행환경과 하위의 물리적 자원들 사이에서 동작하는 계층이다. 이러한 노드 운영체제는 다중의 실행 환경들을 동시에 지원하고, 모든 액티브 노드에 공통된 기능을 제공하고 있다. 이러한 공통 기능은 각 실행 환경이 패킷을 수신 및 전송하기 위한 하위 채널을 구현하고, 다중 환경들에 의한 전송 및 대역폭 계산과 같은 노드 자원들에 대한 액세스를 제어하며, 라우팅과 같은 공통의 서비스를 지원하는 것을 포함한다.

노드 운영체제 인터페이스는 노드 자원에 대한 4가지 추상적 개념을 정의한다. 스레드, 메모리, 채널은 각각 계산, 스토리지, 통신을 위한 시스템 자원들에 대한 개념이다. 플로우에 accounting, 수락 제어를 위한 추상적 개념이며, 시스템 내 자원의 스케줄링을 담당한다. 플로우에 기본적으로 입출력 채널, 메모리 풀, 스레드 풀을 포함한다. 입력 채널에 도착한 액티브 패킷은 해당 플로우에 할당된 스레드와 메모리를 이용하는 실행환경에 의해 처리되고 나서 출력 채널로 전송된다.

각 액티브 노드는 프로그램을 로딩하고 처리하는 방법에 따라 분리 방식(Discrete approach) 방식과 통합 방식(capsule/integrated approach)으로 분류한다[1][2][3][4][6].

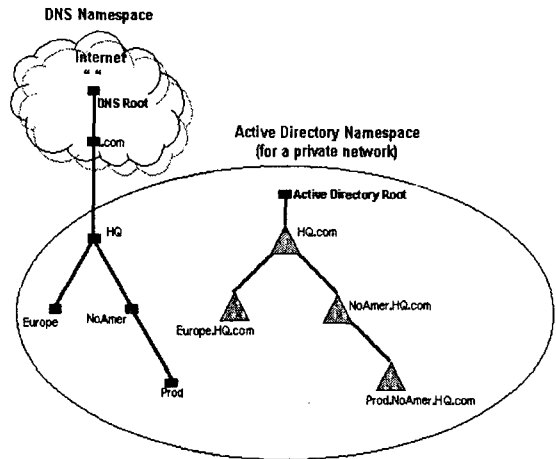
- 분리 방식(discrete approach) : 프로그램이 노드에 상주하여 호스트가 수행하기를 원하는 프로그램 식별자와 필요한 데이터를 저장한 캡슐을 전송하면, 노드는 프로그램 식별자에 해당하는 프로그램을 로딩하여 전달 받은 데이터를 처리하며, 결과를 다음 노드로 전송한다.
- 통합 방식(capsule/integrated approach) : 프로그램이 노드에 미리 저장되어 있지 않는 경우이며, 노드가 프로그램과 데이터를 실은 패킷을 전달한다. 각 노드는 전달 받은 프로그램과 데이터를 자신의 실행 환경에서 수행한 다음 결과를 다음 노드로 전송한다.

3. 액티브 네트워크 관리구조 디자인

현재의 액티브 네트워크 관리 구조는 당분간 전통적인 일반 네트워크 구조와 통합되어 사용되어야 할 것이다. 따라서 기존의 네트워크 관리 시스템과의 효율적 제어를 위하여 액티브 네트워크의 관리 구조 역시 이를 고려한 형태로 구성되어야 하며, 향후 완전

한 액티브 네트워크 구조(Completed Active Network)로 구현되더라도 최소의 관리노력으로 업그레이드 될 수 있도록 하여야 할 것이다[4][6][7]. 따라서 본 논문에서는 효율적인 액티브 네트워크 관리를 위한 Active Network의 Infrastructure를 구성하였다. 본 장에서의 기본 관리구조 단위는 도메인이며, 액티브 네트워크는 기본적으로 도메인으로 구성하였다.

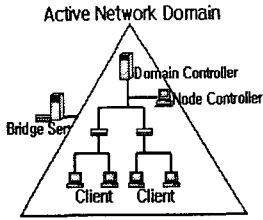
액티브 네트워크에서 주요 고려 사항인 확장성, 보안과 안전, 그리고 관리 효율성을 위하여 관리 단위는 전통적인 Domain Name 방식을 채택하였다[4][5][6][7].



[그림 1] DNS 관리구조

기존 네트워크 시스템에서는 중간노드를 관리 대상(Managed Object)에서 제외 되었으나, 시스템에서 직접 관리하기 위해서는 액티브 노드를 시스템의 관리 대상(Managed Object)으로 편입을 시켰다. 또한 도메인 관리 시스템은 도메인 컨트롤러(DC: Domain Controller)가 수행을 하도록 구성하였으며, 도메인 컨트롤러의 관리 부하를 줄이기 위하여 자신이 관리하는 도메인 영역에 액티브 노드 및 네트워크 자원 관리를 전담하는 전용서버를 구성하였다.

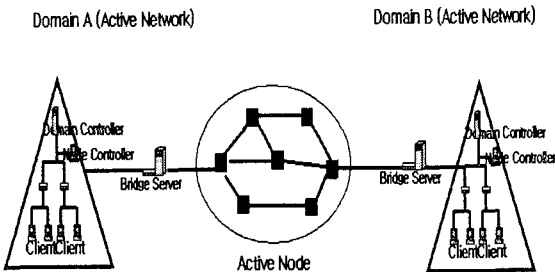
- Domain Controller : 도메인 컨트롤러. 도메인 및 네트워크에 연결되어 있는 네트워크 자원들의 관리를 위한 전용 서버[8].
- Node Controller : 노드 컨트롤러. 네트워크 관리단위(도메인)내에 있는 액티브 노드들을 관리하는 전용 서버 (노드내에 NodeOS 가 없는 일반 노드일 경우는 관리대상에서 제외).
- Bridge Server : 도메인내의 액티브 노드 중 게이트웨이 역할을 하는 서버. 도메인 구성환경에 따라 적절한 캡슐 처리 기능을 수행한다.



[그림 2] 기본 액티브 네트워크 도메인 구조

3.1 Active Node 를 가지는 Completed Active Network Domain

도메인내의 모든 노드가 액티브 노드(Active Node)인 경우이며, 향후 완전한 액티브 네트워크 환경으로 구성되었을 경우이다. 액티브 노드의 프로그램 처리 방식에 따라 2 가지 형태로 액티브 노드가 구성이 될 수는 있으나, 분리 방식과 통합 방식 모두를 수용한 혼합형태의 네트워크 구조를 고려할 수 있다. 혼합 형태의 네트워크 구성에서는 기본적인 노드 수행 프로그램은 액티브 노드에 상주시킨 후, 사용자 또는 서비스 제공자가 필요한 프로그램은 캡슐에 내장하여 노드에서 실행 시킬 수 있다. 이 구조에서는 멀티도메인 상황이라도 도메인과 도메인을 연결하는 노드 역시 액티브 노드인 경우이다. Bridge 서버는 캡슐처리를 액티브 노드와 같은 방식으로 처리한다.



[그림 3] Completed Active Network with Active Node

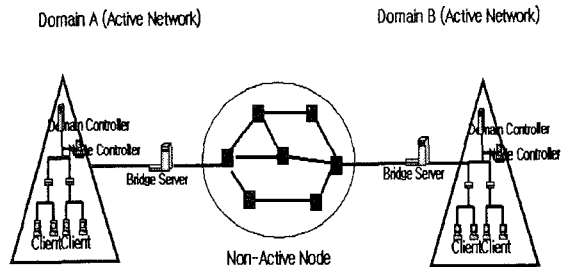
3.2 Non-Active Node 를 가지는 Completed Active Network Domain

이 구조는 2 가지 환경을 가정할 수 있다. 첫째는 도메인 내부의 노드 중 일부가 Non-Active Node 인 경우, 둘째는 도메인과 도메인간의 노드가 Non-Active Node 인 경우이다.

첫번째 상황인 도메인 내부의 노드 중 일부가 Non-Active Node 일 경우는 기존의 액티브 네트워크 연구 분야에서 제시를 하고 있다. 즉, 캡슐이 액티브 노드를 지날 때는 해당 노드에서 필요한 처리를 하지만, 일반 노드(Non-Active Node)를 통과할 때는 forward 시

키도록 구성하고 있다[2][3][4][5][7].

두 번째 상황인 도메인과 도메인 사이의 중간 노드들이 Non-Active Node 인 경우는 좀 더 신중한 디자인이 제시 되어야 한다. 액티브 네트워크의 도메인 구성에서 액티브 노드를 관리 대상으로 편입하였다. 따라서 중간 노드가 액티브 노드 일지라도 자신이 속한 도메인의 관리 제어를 받지 않는 액티브 노드는 보안 정책상 일반 노드로 처리한다(이 가정은 모든 도메인 상황에 적용된다). 도메인 A 에서 도메인 B 로 캡슐이 전송될 때, 도메인 A 의 게이트웨이 역할인 노드(Bridge 서버)는 도메인 A 가 관리하는 마지막 노드가 된다. 도메인 A 에서 마지막 노드를 지나는 순간부터 도메인 B 영역의 게이트웨이 역할의 노드(Bridge 서버)에 도달 하기까지 거치는 중간노드는 모두 일반노드로 인식을 하여 forward 처리만 하도록 한다. 이 경우는 도메인 A 와 도메인 B 는 신뢰 관계(Trust)가 선행되어 있어야 하며, Bridge 서버의 Trust 정보가 각각의 도메인에 소속된 노드 운영체제 컨트롤러 사이에 교환이 되어져야 한다. 도메인의 Bridge 서버에서는 대상 도메인으로 캡슐을 전송하기 전에 Bridge 서버 간에 Trust 관계를 확인하여 이와 같은 캡슐 전송 처리를 한다.



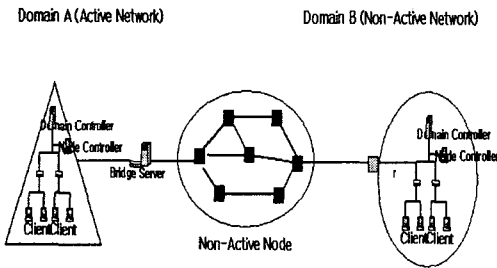
[그림 4] Completed Active Network with Non-Active Node

3.3 Active Network Domain 과 Non-Active Network Domain 간의 도메인 구성

도메인 A 는 액티브 네트워크로 구성되어 있고 도메인 B 는 일반 네트워크로 구성되어 있다고 가정하자. 이 경우 두 도메인 간의 통신에서 중간 노드와 도메인 B 에서의 캡슐이 오히려 네트워크 트래픽과 전송량을 높이는 결과를 낳고 있다. 앞의 가정에서 액티브 네트워크의 도메인 구성에서 액티브 노드를 관리 대상으로 편입하였기 때문에 중간 노드가 액티브 노드 일지라도 자신이 속한 도메인의 관리 제어를 받지 않는 액티브 노드는 보안 정책상 일반 노드로 처리하기로 하였다. 도메인 A 의 Bridge 서버는 이러한 동작을 위하여 액티브 노드의 역할을 서버 수준으로 향상시킨 노드이다.

도메인 A 와 도메인 B 간에 Trust 관계를 맺어서 서로의 네트워크 자원을 공유할 수 있도록 설정하더라도 액티브 네트워크 시스템의 관점에서는 도메인 A

외부로 전송되는 순간부터 캡슐은 일반 네트워크의 패킷처럼 동작을 하게 되어버린다. 이러한 구조에서는 도메인 A의 Bridge 서버는 캡슐을 TCP/IP 기반의 패킷으로 변환하여 네트워크 전송을 하도록 설정하였다. 반대로 도메인 B에서 도메인 A로 패킷이 전송될 때에는 도메인 A의 Bridge 서버는 단지 패킷 forward의 기능만 수행하도록 하였다. 그러나, 도메인 A 내부의 구성 요소들에서 발생한 데이터 통신이나 네트워크 자원간의 통신은 3.1 절의 구조를 따르게 된다.



[그림 5] Active Network Domain with Non-Active Network Domain

4. 결 론

액티브 네트워크는 기존의 패킷 교환 방식의 네트워크 구조가 가지고 있는 비효율적인 부분을 개선하여 사용자에게 의한 계산 및 수행이 가능하도록 시도하는 차세대 네트워크 구조이다. 액티브 네트워크를 통해 이러한 문제를 해결하려는 연구가 여러 연구기관에서 진행 중에 있다. 그러나 현재의 액티브 네트워크 관리에 대한 연구는 관리를 위한 액티브 노드에 중점을 두고 있으나, 네트워크 내의 트래픽을 제어하고 성능을 향상시키며 효율적인 유지, 보수가 용이한 네트워크 구조에 대한 연구는 아직 미흡한 상태이다. 또한 액티브 네트워크는 앞에서 언급한 바와 같이 상당기간 기존의 일반 네트워크와 함께 동작을 할 수밖에 없으며, 표준화된 형태가 정립되어 있지 않기 때문에 호환성과 상호 연동성의 연구과제 역시 아직 남아 있다.

따라서 본 논문에서는 액티브 네트워크 영역에서 기존 네트워크와의 상화 연동 과 다른 액티브 네트워크 구조와의 호환성 및 상호 연동 문제, 그리고 보안 관리에 대한 문제를 해결 하기 위한 기본 구조를 제시 하였으며, 향후 진행되는 액티브 네트워크 관련 연구는 본 논문에서 디자인한 구조를 기반으로 계속 향상시켜 나갈 것이다.

참고문헌

- [1] 이중수, 이승현, 이영희, "Active 네트워크 구조:문제점 및 접근 방향", Sigcomm Reivew, Vol. 1, No. 1, pp109-126, 2000.
- [2] David L, Tennenhouse et al, "A Survey of Active Network Research", IEEE Communication, 1997.
- [3] David L, Tennehouse and Davud J, Wetherall, "Towards and Active Network Architecture", Proc. Multimedia Comp. And Networking, 1996.
- [4] Marcus Brunner, "Active Network and its Management", IEEE, 2000.
- [5] Xu Bin, Qian Depei, Lu Yueming, Wang Lei, "An Active Network -Based Network Management Framework", IEEE, 2000.
- [6] 이종하, 함진호, "Active Network 구조 및 관련 기술", 정보처리학회 제 5 권 제 6 호, 1998.
- [7] Raghupathy Sivakumar, Sung-Wook Han, Vaduvur Bharghvan, "A Scalable Architecture for Active Networks", IEEE, 2000.
- [8] <http://technet.microsoft.com>
- [9] 김현주,장범환,정태명, "Active 네트워크 관리 기술 현황과 전망", Sigcomm Reivew, Vol.1, No.1, pp143-150, 2000.