분산 웹 환경에서 이동 에이전트 보안 모델에 의한 전자서명 메카니즘

최 길 환[†] · 신 민 화[†] · 배 상 현[††]

요 약

현재 이동 에이전트 및 이의 전송과 실행을 위한 이동 에이전트 시스템의 구현에 Java가 많이 사용되고 있지만 Java의 기본적인 보안 모델은 이동 코드의 기능 확장성에 제한을 주는 문제점이 있다. 본 논문에서는 전자서명을 Java 기반의 이동 에이전트에 적용함으로써 시스템의 보안을 유지하면서 이동 에이전트의 기능 확장성을 보장할 수 있음을 보였다. 또한 이동 에이전트의 전자서명과 이의 망 관리로의 적용을 위해서는 망 관리국(NMS : Network Management Station)에서의 서명자 관리나 공개키 관리 등 전자서명과 관련된 기능 외에 망 관리 이동 에이전트의 등록과 전송, 실행 및 실행 관리를 위한 구성이 필요하다. 따라서 이를 위해 작성한 망 관리국과 관리 대상 시스템의 구성 모델을 보였으며, 제안한 구성방식의 동작을 검증하기 위해 망 관리 응용 예를 작성하고 평가하였다. 제안한 구성 방식을 사용하면 전자서명 처리로 인한 속도 저하의 문제가 있지만, 이동 에이전트의 사용으로 인해 얻어지는 부하 분산과 실시간 관리, 망 확장성 증대의 장점 이외에도 관리 기능 및 서비스 추가가 용이한 장점이 있다.

Digital Signature Mechanism by Mobile Agent Security Model of Distributed Web Environment

Kil-Hwan Choi[†] · Min-Hwa Shin[†] · Sang-Hyun Bae[††]

ABSTRACT

Telecommunication networks are becoming bigger and more complex. Its difficult to manage efficiently the networks, because these networks usually have heterogeneous and incompatible components. Nevertheless, current approaches to network management have focused on centralized management strategies based on client-server architecture. These approaches have resulted in much weakness in the real-time management, the service extensibility, and the network scalability. In this paper, we applied the mobile agent technology to solve the above problems. Java is a promising technology for developing mobile agent system. But, there are several problems like the service extensibility in using the Java. To solve these problems, a new approach using digital signature is suggested to authenticate mobile agent in network management environments. This approach can solve the conflict between security of the system and extensibility of the mobile code. Moreover, the system suggested in this paper show the decentralized and flexible network management solutions.

키워드 : 이동 에이전트(Mobile Agent), 보안(Security), 전자서명(Digital Signature)

1. 서 론

신속하고 다양한 정보통신 서비스에 의존하는 정보화 사회로의 진행이 가속되면서 유무선망의 구분 없는 개인 통신 서비스나 음성과 데이터 및 화상정보가 통합되는 새로운 통신 서비스가 요구되고 있다. 이러한 서비스를 지원하기 위한 통신망의 규모가 커지고 복잡성이 증가하고 있으며, 망 요소간 또는 망간 이질성이 증대됨에 따라 망 요소들의 효율적인 관리가 필요하다.

현재까지 사용되는 대부분의 모델은 분산 시스템의 클라이언트/서버(Client/Server) 구조에 기반을 둔 매니저/에이전트(Manager/Agent) 구조를 채택하고 있다. 이는 관리 기능을 가지고 있는 NMS(Network Management Station)와 관리 대상인 NE(Network Element)들이 망 관리 프로토콜(Network Management Protocol)을 이용하여 망 관리에 필요한 정보를 교환하고 처리하는 구조를 가진 중앙집중형(Centralized Network Management) 모델이다[2]. 인터넷의 망 관리 표준인 IETF의 SNMP(Simple Network Management Protocol[3])와 OSI에서의 망 관리 표준인 CMIP(Common

† 준 회 원 : 조선대학교 대학원 전산통계학과
†† 종신회원 : 조선대학교 전산통계학과 교수
논문접수 : 2003년 7월 22일, 심사완료 : 2003년 8월 23일

Management Information Protocol[4])에서 이 모델이 사용되고 있다.

중앙집중형 망 관리 모델은 망 구성 및 관리가 단순하고 보안이 용이하다. 그러나, NMS에 모든 관리 기능이 집중되기 때문에 망 확장성과 실시간 관리 기능이 제한되고, 서비스의 동적인 추가와 망 통합관리가 어렵다는 단점이 있다[5].

이와 같은 중앙집중형 망 관리 모델의 문제점을 보완하기 위하여 망 관리에 이동 에이전트(Mobile Agent) 기술을 적용하려는 연구가 진행중이다. 이는 이동 코드(Mobile Code) 또는 이동 에이전트라 불리는 실행 가능한 코드 자체가 네트워을 통해 해당 시스템으로 전송되어 실행되는 방식으로서, 현재 대부분의 분산 환경을 이루고 있는 클라이언트/서버 기술인 RPC(Remote Procedure Call)나 메시지 교환과는 대조되는 개념이다[6].

Mobile Agent는 플랫폼의 독립적인 분산 환경에서의 프로그래밍을 기존의 RPC, 메시지 전달, 자바와 같은 언어의 코드 이동 환경을 제공한다. 또한, Mobile Agent는 자율성을 가지고 사용자나 조직의 권한을 대행하여 특정 기간 동안 혼자 독립적으로 수행될 수도 있고, 또는 사회성을 가지고 다른 에이전트와의 상호 교류를 통하여 보다 복합적인 기능을 수행할 수 있다. Mobile Agent는 이외에도 반응성, 이동성 등과 같은 속성을 가지는 자율적인 프로그램이다[13, 14].

이동 에이전트 기술을 망 관리에 적용하면 관리 기능을 NE들로 분산시킴으로써 중앙집중형 관리 모델에서 발생되는 NMS의 부하 집중 문제를 해결할 수 있다. 또한 관리 작업의 결과가 각 NE에서 처리되어 NMS로 전송되므로 NMS와 NE간의 빈번한 정보교환으로 인하여 발생하는 네트웍 부하를 감소시킨다. 또한, 관리 기능을 동적으로 추가하는 것이 용이하며, 이기종 망간의 관리 연동 등의 장점이 있다[7, 8].

본 논문의 구성은 다음과 같다.

제 2장에서는 웹 기반의 이동 에이전트 시스템 및 Mobile Agent를 구성하면서 생기는 문제 및 해결 방안을 살펴본다. 이동 에이전트 시스템 개발에 많이 사용되는 Java의 보안 모델과 전자서명(Digital Signature)을 이동 에이전트에 적용함으로써 시스템 보안 문제와 이동 에이전트의 기능 확장성 제한 문제를 해결하였다. 제 3장에서는 제안한 망 관리 이동 에이전트 실행 환경 및 망 관리국의 구성 방법을 보였다. 제 4장에서는 제안한 방식을 이용한 망 관리 응용을 통한 이동 에이전트 실행환경의 동작을 검증하였다. 또한 응용의 성능 평가를 통해 이동 에이전트를 사용한 망 관리 모델의 장점을 확인하였다. 마지막 제 5장에서는 결론과 향후 연구과제에 대해 기술한다.

2. 웹 기반의 이동 에이전트 시스템

이동 에이전트가 네트워을 이동하면서 기능을 수행하기 위해서는 각 플랫폼에 에이전트 실행 엔진(Agent Execution Engine) 또는 에이전트 실행 환경(Agent Execution Environment)이라 불리우는 이동 에이전트를 실행시키는 기능을 가진 모듈이 필요하다.

객체 지향적이고 분산 환경에 적합한 네트웍 언어이며 플랫폼에 무관한 이식성 등의 특성 때문에 최근들어 이동 에이전트 시스템 구현에 Java가 많이 사용되고 있다[1, 8, 9].

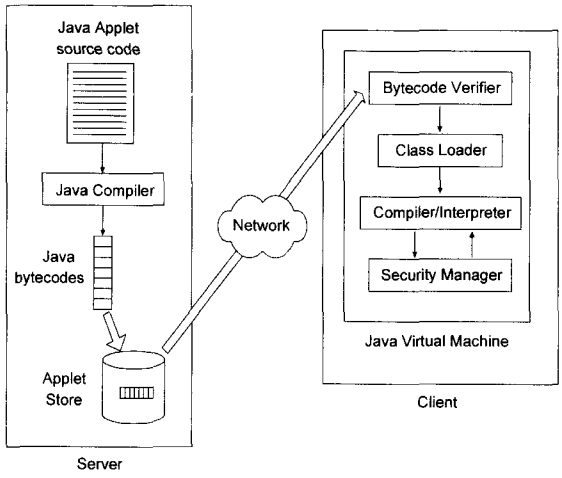
본 논문에서도 Java 언어를 이용하여 이동 에이전트 및 실행환경을 구현하였다. 다음에 이동 에이전트와 이의 실행환경 구성에 Java를 사용할때 Java의 보안 모델로 인해 발생되는 문제점을 기술하였다.

2.1 Java 보안 모델

Java 코드는 분산환경에서 수행되도록 설계되었다. 따라서 네트웍을 통해서 전달된 Java 코드를 수행하는 클라이언트 시스템의 안전성을 보장하는 것은 중요한 사항이다. 특히 이동 에이전트는 네트웍을 통해서 플랫폼간을 이동하면서 실행되기 때문에 적절한 제약이 가해지지 않는다면 시스템의 보안에 많은 문제를 야기시킬 수 있다. 이동 에이전트를 악의적인 호스트에서 보호하기 위해서는 다음과 같은 보안 정책이 필요하다[15-17].

첫째, 호스트 컴퓨터의 인증 서비스는 호스트의 사칭 위협을 방지하기 위한 것으로 이동 에이전트를 생성한 호스트와 이동할 호스트 사이에서 서로의 컴퓨터가 서로 신뢰성 있는 호스트임을 인증해야 한다. 둘째는 에이전트 코드와 데이터의 기밀성으로 이동 에이전트의 코드와 데이터가 네트워크 상에서 제삼자에게 스누핑되어 데이터가 누설되는 것을 방지하기 위해 데이터를 암호화하여 전송한다. 셋째는 에이전트 코드와 데이터의 무결성으로 이동 에이전트의 코드와 데이터가 네트워크 상에서 제삼자에 의해 변형되지 않고 제대로 전송되었는지 확인하는 서비스이다. 넷째는 에이전트 전송의 부인 방지로 호스트 컴퓨터가 이동 에이전트를 보내고 받는 것에 대한 부인을 할 수 없도록 하는 서비스이다. 그리고 마지막으로 이동 에이전트 실행 감사 기능으로 이동 에이전트는 코드와 데이터로 구성되어 있다. 에이전트의 코드는 생성된 후 호스트 컴퓨터를 이동하는 도중에 변하지 않지만 에이전트 실행 데이터는 항상 변화한다. 문제는 호스트 컴퓨터가 에이전트의 실행 데이터를 불법적으로 수정하는데 있다. 호스트 컴퓨터들을 이동하면서 얻은 정보들이 한 호스트에서 모두 삭제되거나 수정되어 에이전트의 행동에 영향을 줄 수 있다. 이를 탐지하기 위해 에이전트 실행 데이터 변화에 대한 감사(auditing) 기능을 제공한다.

(그림 1)에 Java 애플릿(Applet)의 생성과 전달, 보안사항의 적용 및 수행 과정을 나타내었다.



(그림 1) Java 코드의 생성과 전달, Java 보안 모델

2.2 전자서명과 Signed Applet

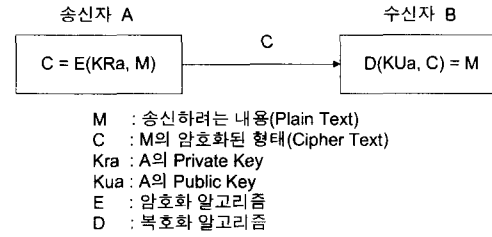
시스템의 보안을 만족시키고 이동 에이전트 기능 확장의 어려움을 극복하기 위해 본 논문에서는 전자서명(Digital Signature) 기술을 이용한다. 망 관리 이동 에이전트를 전자서명하여 전송하면, 이동 에이전트 엔진이 있는 관리 대상 시스템에서 인증 확인 작업을 거친 후 해당 에이전트에 대해 파일 시스템과 네트워 접근 제한을 해제하는 것이다. 이러한 방식을 사용하면 보안을 만족시키면서, 망 관리 이동 에이전트의 기능 제한을 극복할 수 있다.

2.2.1 전자서명

전자서명(Digital Signature)은 공개키(Public Key) 방식을 사용하여 사용자 인증과 메시지 불변성을 보장 해주는 기술이다. 공개키 방식이란 사용자마다 자신만이 알고 있는 Private Key와 이에 대응되면서 다른 사람들에게 알려줘야 하는 Public key, 한 쌍의 키들을 암호화와 복호화에 사용하는 것이다. 이 때 Private Key를 사용하여 메시지를 암호화 한 경우 Public Key를 사용하여야 해독할 수 있고, 반대로 Public Key를 사용하여 암호화 한 경우는 대응되는 Private Key를 사용하여야 해독 가능하다.

전송하고자 하는 상대의 Public Key로 메시지를 암호화 한 경우에는 원하는 상대만이 자신(수신자)의 Private Key를 사용하여 해독할 수 있으므로 메시지 내용의 기밀성이 필요한 경우에 사용된다.

반대의 경우, 즉 자신의 Private Key로 암호화 한 경우에는 공개되는 Public Key로 누구나 해독할 수 있으므로 내용의 기밀성 보다는 메시지의 작성자를 인증하고 메시지 내용의 불변성을 인증하는 데에 사용되며, 이를 전자 서명이라 한다. (그림 2)에서는 송신자 A가 자신의 Private Key (KRa)를 사용하여 암호화 알고리즘 E를 통해 원문 M을 암호화된 형태인 C(Cyphertext)로 변환 후 전송한 것을 나타낸다.

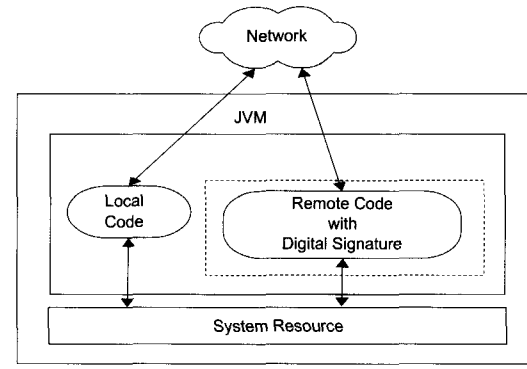


(그림 2) 전자서명 메커니즘

2.2.2 Signed-Applet

Java 모델에서는 네트웍을 통해 이동 가능한 Java 코드를 애플릿(Applet)이라 하며, 이에 대해 앞에서 언급한 Sandbox라 불리는 제한이 적용된다. Signed-Applet이란 JDK 1.1부터 지원되는 것으로써 전자서명을 사용하여 인증된 애플릿을 의미한다. 전자서명된 Java 애플릿은 이동 후 실행 환경에서 검증과정을 거친 후 Sandbox의 제한을 벗어나서, 마치 Local에서 실행되는 것처럼 파일 시스템과 네트웍 액세스를 자유로이 할 수 있다[10].

Sandbox의 제한에서 벗어나서 실행되는 Signed-Applet을 (그림 3)에서 보였다.



(그림 3) Signed Applet의 실행

2.3 Mobile Agent를 구성하면서 생기는 문제점 및 해결 방안

MA는 자의적으로 네트워크를 돌아다니면서 자신의 목표를 수행하는 프로그램이다. 따라서 이 MA에서 발생하는 문제는 MA를 받아들여 실행시켜주는 호스트 쪽의 문제가 있을 수 있고, 네트워크를 돌아다니면서 수행하는 MA 자체의 문제로 나누어 볼 수 있다[12].

첫째, 호스트는 MA를 받아들여서 실행환경에서 MA가 요청한 서비스를 수행하고, MA에 결과를 반환해야 한다. 하지만 호스트는 받아들이는 MA의 특성을 수행하기 전까지는 알아 볼 수가 없다. 따라서 받아들인 MA가 Virus나 Worm과 같은 행동을 하는 경우에, 그 MA로부터 호스트 자신을 보호해야 하고, 또 Denial of Service와 같이 호스트의 컴퓨팅 파워를 무력화시킬 수 있는 MA로부터 보호해야 한다. 또한 MA가 호스트의 권한을 얻어서 호스트의 리소스와 데이터를 고갈시키거나 마음대로 접근하는 것으로부터 보호

해야 한다. 둘째, MA는 MA의 수행코드를 호스트의 수행 환경에게 수행을 요청하고 그 결과를 가지고 다른 호스트로 이동을 하던지 그 데이터를 사용자에게 반환해야 한다. 하지만, 악의적인 호스트는 MA의 코드를 마음대로 바꾸어 수행하여, 결과를 조작할 수 있고, MA가 수집한 데이터를 마음대로 조작할 수 있다. 따라서 MA의 코드와 수집한 데이터나 수행 결과를 악의적인 호스트가 마음대로 변경시키지 못하게 해야 하며, MA가 수집한 데이터를 노출시키지 말아야 한다. MA는 호스트뿐만 아니라 다른 MA와 통신을 통해서 다른 MA가 수집한 결과를 얻을 수 있다. 이때, MA가 다른 MA와 통신하기 위해서 KQML를 사용할 수 있다. 이 경우, 악의를 가진 MA나 MA를 무력화시키기 위해서 임의로 생성한 MA의 복제물들이 MA를 공격할 수 있으므로, 다른 MA로부터 자신을 보호할 수 있어야 한다[11].

이러한 문제점을 해결하기 위해서는 첫째, MA안에 있는 코드의 작성자의 존재는 작성자가 code에 대한 Sign을 한다면 쉽게 결정될 수 있다. 같은 방법으로 MA의 송신자 또한 Sign의 방법으로 송신자의 존재를 쉽게 판별할 수 있다. 둘째, 작성자의 Signature를 점검해 봄으로써 쉽게 MA 코드의 무결성을 검사할 수 있다. 셋째, 호스트간에 MA를 전송시키는 동안 암호화 방법을 쓴다면, 권한이 없는 파티가 MA가 가지고 있는 중요한 정보를 읽는 것을 막을 수 있다. 넷째, 인터프리터는 MA가 MA의 작성자, 프로그램, 사용자 그리고 상태를 고려하여 리소스에 접근의 허용을 결정할 수 있다.

2.4 Mobile Agent의 보호

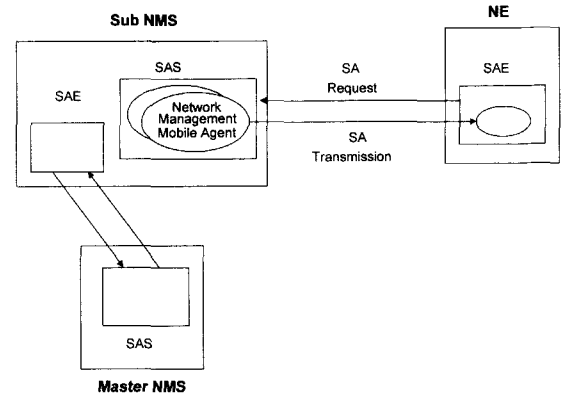
특정 작업을 성취하기 위해, 큰 하나의 문제를 여러 개의 작은 목적 혹은 특정한 작업으로 나눈 후 그 작업을 MA가 수행하도록 작업별로 나뉘어진 여러 개의 MA를 네트워크로 전송하였을 경우에는, 호스트와는 달리 작업별로 나뉘어진 MA는 주어진 목적을 성취하기 위해서 여러 호스트 사이를 이동하며, 결국엔 나뉘어진 MA가 수집한 결과를 모았을 경우, 전체 작업에 대한 결과를 얻을 수 있다. 이러한 check-and-balance 접근방법에서 그 어떤 MA도 큰 문제의 전체적인 요소를 수집할 수 없고, 전체적인 결과를 소유하지 못한다. 따라서 필요한 정보를 여러 개의 작은 조각으로 나누어 각각의 조각을 MA로 나누어 실행하면, 특정한 호스트는 전체 정보를 알아볼 수가 없으며, 이를 통해 MA가 의도하고자 하는 행동 혹은 MA가 모아온 데이터를 효율적으로 분산시켜, MA를 보호할 수 있다.

3. 망 관리 이동 에이전트 시스템

3.1 전자 서명된 망 관리 응용 이동 에이전트 구성

본 논문에서는 망 관리 기능을 가지며 전자 서명된 이동 에이전트를 SA(Signed Agent)라 한다. SA의 유지, 전달 및 실행을 위해 SAS(Signed Agent Server)와 SAE(Signed Agent Executor), 두 개의 기능 모듈을 구성하였다.

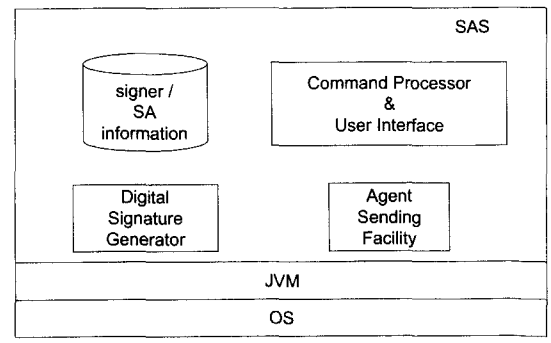
(그림 4)에서 Master NMS 자신은 관리 대상이 아니므로 망 관리 SA들을 유지하는 SAS만을 수행하며, NE는 관리 대상으로서의 역할만을 하므로 SAE만을 실행한다. Sub Master는 자신이 NMS의 역할을 수행하면서 또한 관리 대상일 수 있으므로 SAE와 SAS를 둘 다 실행하고 있다.



(그림 4) 이동 에이전트 시스템 구성

3.2 SAS(Signed Agent Server)

SAS의 기능 모듈을 (그림 5)과 같다. SA를 SAE에 전송하는 기능을 수행하는 Agent Sending Facility와 전자서명을 생성하는 Digital Signature Generator를 사용하여 Command Processor는 서명자 관리, SA 정보 관리, Signed-Agent 수행 관리를 제공한다.

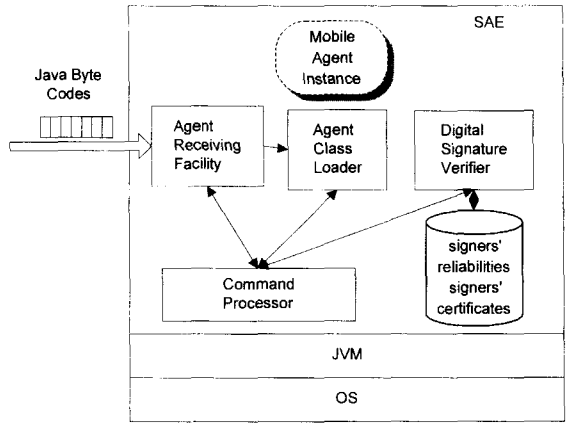


(그림 5) SAS의 구성

3.3 SAE(Signed Agent Executor)

SAE는 관리 대상에서 Daemon 형태로 동작하면서 자신이 요구했거나, SAS에서 전달한 SA에 대해 전자서명을 검사하고 실행시키는 역할을 담당한다. SAE는 관리 대상에서 실행되므로 사용자 인터페이스는 가지지 않는다. 또한 현재 구현된 SAE는 전자서명된 이동 에이전트를 수신하여 실행시키는 역할만을 하며, 코드 이동이나 에이전트간 통신은 지원

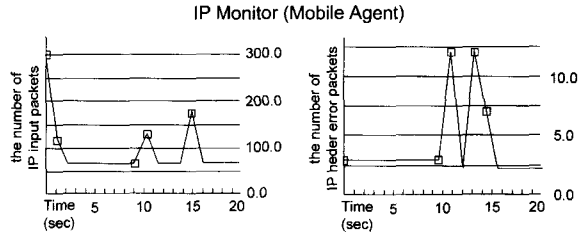
하지 않는다. (그림 6) SAE의 구성 모듈을 나타내었다.



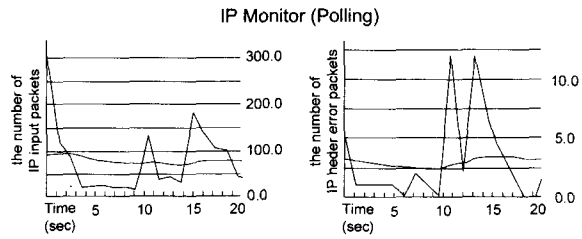
(그림 6) SAE의 구성

4. 실험 결과 및 평가

(그림 7)(a) 서버 응용이 표시하는 화면이다. 그래프의 X축은 한 눈금당 1초씩 시간의 경과를 나타내며, Y축은 해당 시점에서의 각 MIB 값의 증가 값, 즉 단위 시간(1초)당 IP 수신 패킷의 수와 오류 패킷의 수를 나타낸다. 서버 응용은 최근 20초 동안에 클라이언트 응용에게서 ipInReceives 값에 대해 5개의 메시지 이벤트 4건, 평균 값 1건을, ipInHdrErrors 값에 대해서는 5개의 메시지 이벤트 3건, 평균 값 2건을 수신했음을 알 수 있다. 따라서 20초 동안에 총 10개의 메시지가 교환되었다.



(a) 이동 에이전트 방식



(b) Polling 방식

(그림 7) IP 상태 감시 응용 결과

(그림 7)(b) 비교를 위해 전통적인 SNMP 매니저/에이전트 구조에서 Polling 방식을 사용한 실험 결과이다. 이 방식에서 매니저는 1초에 한 번씩 ipInReceives와 ipInHdrErrors 값을 요구하는 SNMP PDU들을 SNMP 에이전트에게 전달하고, 관리 대상에 있는 SNMP 에이전트는 이에 응답하게 된다. 매니저 프로그램은 수신한 MIB 값들과 평균 값을 표시한다. 이 방식에서는 1초당 2개의 요구 SNMP PDU들과 응답 SNMP PDU들이 발생하게 된다. 따라서 20초 동안에 총 80개의 SNMP PDU가 발생하게 된다.

Polling 간격이나 평균 값의 보고 간격 및 이벤트의 정의에 따라 차이가 있겠지만, 이동 에이전트 모델에서는 관리기능의 많은 부분을 관리 대상에서 수행하므로 실시간 관리가 가능하다. 또한 관리 정보에 대한 처리가 NE에서 수행되어서 추상화된 정보의 형태로 NMS로 전달되기 때문에 NMS의 부하를 분산시키고 교환하는 정보의 양을 줄일 수 있다.

5. 결론 및 향후 과제

일반적인 망 관리 모델은 클라이언트/서버 구조에 근거한 중앙집중형으로서, 망 확장성이나 실시간 관리 및 동적인 서비스 추가에 있어서 단점을 가지고 있다. 또한 Mobile Agent 시스템에서 보안문제는 그 시스템의 유용성만큼이나 다양하고 해결하기 난해한 문제가 많이 존재한다. MA로부터 호스트를 보호하는 것이나 호스트로부터 MA를 보호하는 것은 둘다 매우 중요하고 쉽지 않은 문제이다. 따라서 필요한 정보를 여러 개의 작은 조각으로 나누어 각각의 조각을 MA로 나누어 실행하면, 특정한 호스트는 전체 정보를 알아볼 수가 없으며, 이를 통해 MA가 의도하고자 하는 행동 혹은 MA가 모아온 데이터를 효율적으로 분산시켜, MA를 보호할 수 있는 장점이 있다.

본 논문에서는 전자서명된 망 관리 이동 에이전트를 위한 시스템을 구성함으로써 보안을 유지하고, 망 관리 응용의 기능 제한을 극복했다. 또한 망 관리 이동 에이전트의 동작 실례를 보이고 성능을 평가함으로써 제안한 시스템의 동작을 검증하고, 이동 에이전트를 이용한 망 관리 모델의 장점을 보였다.

현재에는 망 관리를 위한 이동 에이전트를 일반적인 Java 애플릿 형태로 구성하였지만, 망 관리에 적합한 형태의 망 관리 이동 에이전트의 구조를 고안하는 것이 바람직 할 것이다. 또한 전자서명을 하기 위해서는 어떤 프로토콜을 적용할 것인가를 연구하여 결정해야 한다. 이러한 프로토콜에는 다른 매개 수단을 통하지 않고 송·수신자간에 직접 메시지와 서명을 송·수신하는 직접서명 방식과 송·수신자간에 제 3자의 중재자를 통하여 메시지를 송·수신함으로써 중재자에 의해서 송신자의 서명이 인증되는 간접서명 방식이 있고 직접 서명방식과 간접 서명 방식을 모두 사용하는 혼합 서명 방법이 있다. 그리고 전자서명을 활용하기 위해서는 메시지의 크기와 용도에 따라서 서명을 어떻게 할 것인가를 결정해야 한다 점이 대두된다. 또한 망 관리 이동

에이전트의 실행 엔진 역할을 하는 SAE는 이동 에이전트 코드 및 이동 에이전트가 수집한 데이터를 다른 곳으로 이동시킬 수 있는 기능과 에이전트간 통신 기능이 구현되야 하며, 이 역시 망 관리에 적합한 형태로 설계되어야 한다.

참 고 문 헌

[1] 최길환, 배상현, 자바 기반의 이동 에이전트 보안 구조 설계와 암호기능 구현, 한국인터넷정보학회논문집, Vol.3, No.1, pp. 61-69, 2002.

[2] Roch H. Glitho and Stephen Hayes, Telecommunications Management Network : Vision vs. Reality, IEEE Communication Magazine, March, 1995.

[3] J. Case, M. Schoffstall, J. Davin, "A Simple Network Management Protocol (SNMP)," RFC 1157, May, 1990.

[4] Y. Yemini, The OSI Network Management Model, IEEE Communication Magazine, May, 1993.

[5] German Goldszmidt, Yechiam Yemini, "Distributed Management by Delegation," In the 15th International Conference on Distributed Systems, June, 1995.

[6] Colin G. Harrison, David M. Chess and Aaron Kershenbaum, "Mobile Agents : Are they a good idea?," IBM Research Report RC 19887, Available From http://www.research.ibm.com/xv-953-mobag-ps, 1995.

[7] M. Baldi, S. Gai and G. P. Picco, "Exploiting Code Mobility in Decentralized and Flexible Network Management, In the first International Workshop on Mobile Agents (MA '97), April, 1997.

[8] Bieszczad, A. and Pagurek, B., Towards plug-and-play networks with mobile code, In the International Conference for Computer Communications 97 (ICCC '97), pp.19-21, November, 1997.

[9] K. Kotay, D. Kotz, Transportable Agents, In the Third International Conference on Information and Knowledge Management (CIKM '94), December, 1994.

[10] Sun Microsystems co., Security and Signed Applets, Available From http://java.sun.com/products/jdk/1.1/docs/guide/security/index.html

[11] Jonathan T. Moore, Mobile Code Security Techniques, Available from authors, May, 1998.

[12] William M. Farmer, Joshua D. Guttman, and Vipin Swarup, Security for Mobile Agents : Issues and Requirements, Available from authors, 1998.

[13] Bennet S. Yee, A sanctuary for Mobile Agents, Available from authors, February, 1997.

[14] Tomas Sander and Christian F. Tschudin, Protecting Mobile Agents against Malicious Hosts, Available from authors, November, 1997.

[15] Colin G. Harrison, Mobile Agents : Are they a good idea?, Available from authors, March, 1995.

[16] Danny B. Lange and Mitsuru osima, Programming and Deploying Java Mobile Agents with Aglets, 1998.

[17] Giovanni Vigna, Protecting Mobile Agents through Tracing, Available from authors, 1998.

최 길 환

e-mail : ckhplc@hanmir.com

1992년 한밭대학교 전기공학과(공학사)

2000년 조선대학교 일반대학원 전산통계학과(이학석사)

2003년 조선대학교 일반대학원 전산통계학과 박사수료

관심분야 : 이동 에이전트, 분산 처리, 보안, 네트워크

신 민 화

e-mail : minanddjih@hanmail.net

1992년 조선대학교 전자공학과(공학사)

2000년 목포대학교 전산통계학과(이학석사)

2003년~현재 조선대학교 전산통계학과 박사과정

관심분야 : 인공지능 및 컴퓨터응용분야

배 상 현

e-mail : shbae@mail.chosun.ac.kr

1982년 조선대학교 전기공학과(공학사)

1984년 조선대학교 대학원 전기·전자공학과(공학석사)

1985년 일본동경공대 전자정보통신공학연구소 연구원

1988년 일본동경도립대학 정보공학과(공학박사)

1995년 일본과학기술원 전자정보통신공학부 초빙교수

1998년 일본동경도 멀티미디어 연구소 초빙교수

1988년~현재 조선대학교 자연과학대학 전산통계학과 교수

관심분야 : 대규모지식베이스, 인공신경망, 퍼지시스템