
분할관리정책에 의한 이동 에이전트를 이용한 효율적인 네트워크 관리

Effective Network Management Using Mobile Agents Based on Split Management Policy

이재형, 오길호
금오공과대학교 컴퓨터공학부

Jae-Hyoung Lee(zzeng09@kumoh.ac.kr), Gil-Ho Oh(gilho@kumoh.ac.kr)

요약

전통적인 SNMP를 이용한 중앙집중형 네트워크 관리 방식은 빈번한 폴링에 의한 통신량의 증가와 특정 관리 노드에 병목 현상이 생기는 문제점을 안고 있다. 이러한 문제를 해결하고 극복하기 위한 이동 에이전트에 의한 네트워크 관리방식들이 연구되어 오고 있다. 그러나 이동 에이전트에 의한 네트워크 관리 방식은 관리해야 될 노드가 증가함에 따라 이동되는 데이터량이 누적 증가되어 효율성이 떨어진다. 이러한 문제점을 해결하기 위하여 본 논문에서는 이동에이전트를 이용한 분할관리 방식을 제안한다. 제안된 방법에서는 노드들은 그룹들로 나누어지고 별도의 이동에이전트들에 의하여 관리된다. 각각의 이동에이전트들은 자신의 노드들을 방문하고 정보를 수집한다. 제안된 방법이 효율적임을 실험을 통하여 보였다.

■ 중심어 : | 이동 에이전트 | 네트워크 관리 |

Abstract

The traditional centralized network management method using the SNMP has problems, such as, increase of communication traffic and severe bottleneck phenomena at the specific manager-node caused by a frequent polling mechanism. Several mobile agent based network management methods have been studied to solve and overcome these problem. However, in these mobile agent based methods, as the number of nodes to be managed is increased, the mobile data are increased accumulatively and the effectiveness of the network management is decreased. In this paper, we propose the split management method by using mobile agents to handle these problems. In our method, the nodes are divided into groups to be handled by separate mobile agents. Each mobile agent visits its own nodes and gets management information from them. We demonstrate the effectiveness of our proposed method using experimental results.

■ keyword : | Mobile Agent | Network Management |

I. 서론

네트워크의 대형화, 복잡화, 네트워크를 이용한 업무

증가 등으로 네트워크 관리에 대한 중요성이 부각됨으로 이에 따른 통신망 관리의 상호 운용성 제공을 목적으로 IETF는 SNMP(Simple Network Management

* 본 연구는 금오공과대학교 교내학술연구비의 지원으로 연구되었습니다.

접수번호 : #070704-003

접수일자 : 2007년 07월 04일

심사완료일 : 2007년 08월 22일

교신저자 : 오길호, e-mail : gilho@kumoh.ac.kr

Protocol)[1], ISO/OSI는 CMIP(Common Management Information Protocol)[2] 표준을 개발하였다. 이중 SNMP는 TCP/IP에 기반한 RPC(Remote Procedure Call) 방식의 네트워크 관리 프로토콜로서, 구조가 단순하며 구현도 용이하다는 장점으로 현재 가장 널리 사용되고 있는 프로토콜로 자리 잡았다. 그러나 중앙 집중형 관리의 문제점으로 인해 이동 에이전트를 이용한 네트워크 관리에 대한 연구가 계속되고 있으며 이는 중앙 집중형 방식보다 효율적임이 증명되고 있다. 그렇지만 기존의 이동 에이전트를 이용한 네트워크 관리방식에서는 에이전트의 이동성이라는 특성으로 인한 원격 통신비용을 줄일 수 있다는 장점은 부각되었지만, 노드 순회시 획득한 데이터들이 누적되기 때문에 오히려 네트워크 통신비용이 증가할 수 있다는 문제점들은 크게 고려하지 않았다[3-6]. 따라서 본 논문에서는 노드 순회시 이동 에이전트의 누적되는 데이터가 전체 통신비용에 미치는 영향을 고려하여, 방문해야할 전체 노드 중에서 통신비용을 최소화 할 수 있는 최적의 분할 영역을 결정하고, 전체 노드를 결정된 분할 영역으로 나누어 관리함으로써 통신비용의 최적화를 위한 분할 관리 정책을 제시한다.

II. 네트워크 관리를 위한 중앙 집중형 모델과 이동 에이전트 모델 비교

전통적인 네트워크 관리 모델인 중앙 집중형 네트워크 관리 모델과 이동 에이전트를 기반으로 한 네트워크 관리 모델의 특성을 서로 비교한다.

1. 중앙 집중형 네트워크 관리 모델

중앙 집중형 관리 모델인 SNMP는 관리자와 에이전트 사이의 통신을 통한 네트워크 관리를 수행하게 되는 구조로서, 관리자는 에이전트에게 관리에 필요한 특정 관리객체의 값을 읽어오는 Get(get-next)과 객체의 값을 변경하는 Set을 수행하며 에이전트는 관리자에게 특정 상황 발생시 Trap을 수행한다. SNMP를 이용한 네트워크 관리는 구현이 용이하고 관리가 간편하다는 장

점이 있는 반면, 하나의 중앙 관리자가 관리 대상이 되는 모든 SNMP 에이전트와 관리 정보를 주고받기 때문에, 관리자에게 상대적인 부하의 증가가 발생하게 된다. 따라서 일정 수준 이상의 관리 대상이 늘어나게 될 경우 관리자의 성능뿐 아니라 네트워크 통신량의 급격한 증가를 보인다. 네트워크의 실제적인 관리는 폴링(Polling)을 통한 관리 요청이 관리 대상으로부터 중앙의 관리자에게 전달된 후에 이루어지게 되므로 문제 발생시 즉시 대처할 수 있는 실시간 관리가 어렵게 된다 [7][8]. 또한 관리 서비스 기능이 사전에 정의됨으로 새로운 관리 서비스의 추가가 어렵다. 따라서 새로운 네트워크 관리 요소의 도입이 필요할 경우 네트워크 관리 시스템은 관리 요소의 도입을 수용할 수 있을지 고려해야 한다. [그림 1]은 중앙 집중형 네트워크 관리방식을 간략하게 나타낸 것이다.

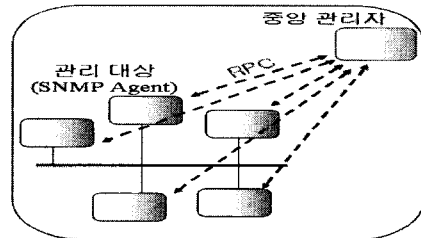


그림 1. 중앙 집중형 네트워크 관리 방식

2. 이동 에이전트를 이용한 네트워크 관리 모델

중앙 집중형 관리의 문제점에 대한 방안으로 이동 에이전트를 네트워크 관리에 적용한 연구가 계속되고 있다[6][9]. 이동 에이전트 기술은 기존의 프로세스간 통신(Remote Procedure Call)을 사용한 네트워크 프로그래밍 개념의 확장으로 호스트간의 원격 상호 작용 수를 줄여 네트워크 부하를 감소시킬 수 있다. 이는 실행 가능한 코드가 자신의 상태와 정보를 포함하여 네트워크를 통해 관리 대상에 이동함으로써 관리 작업을 지역적으로 수행하게 된다[10]. [그림 2]는 이동 에이전트를 이용한 네트워크 관리방식을 나타내고 있다. 네트워크 관리를 위한 기본 구성요소들은 다음과 같다.

• 관리자 (Manager)

이동 에이전트를 생성 및 파견하고 관리 노드를 순회하고 돌아오는 이동 에이전트의 결과를 받아 결과를 저장 및 출력하는 역할을 담당한다.

• 이동 에이전트 (Mobile Agent)

관리 대상을 순회하며 자신에게 할당된 관리 작업을 수행하므로 각 관리 요소들의 관리 정보를 회수하며 자신에게 주어진 역할을 담당한다.

• 이동 에이전트 서버(Mobile Agent Server)

이동 에이전트를 받아들이고 자신의 관리 자원에 접근할 수 있는 인터페이스를 제공해주며 악의적인 코드를 방지하는 보안 관리자 등을 담당하므로 이동 에이전트가 네트워크 관리 작업을 수행하는데 필요한 기능들을 제공하는 역할을 한다.

• 이동 에이전트 생성자(Mobile Agents Generator)

각각의 네트워크 관리 기능에 필요한 이동 에이전트의 생성 요청시 각 서비스에 맞는 이동 에이전트를 생성하는 역할을 담당한다.

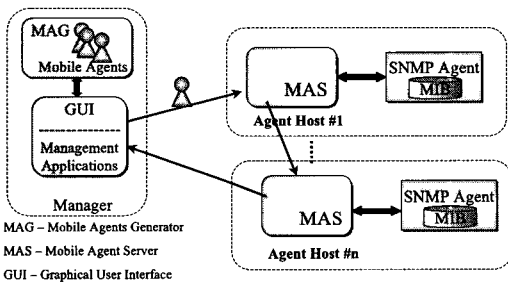


그림 2. 이동 에이전트 기반의 네트워크 관리 구조

이러한 이동 에이전트를 이용한 방식이 중앙집중형 네트워크 관리방식에 비하여 네트워크 관리를 위한 요구와 응답 메시지를 줄여 네트워크 오버헤드를 감소시키고 부하를 분산시키는 효과를 가져 오고 네트워크 관리자의 부하가 네트워크의 크기에 큰 영향을 받지 않기 때문에, 네트워크 관리의 범위 확장(Scalability)을 비교

적 용이하게 할 수 있다. 실시간성이 강하게 요구되는 네트워크 관리 동작일 경우 이동 에이전트 방식은 직접 그 곳에서 수행되기 때문에 빠르게 대처할 수 있다. 그리고 각각의 구조와 관리 기능에 따른 관리 대상에 대해 해당 관리 서비스를 가지고 있는 이동 에이전트를 전송시켜, 관리 대상을 관리함으로써 다양한 서비스의 추가가 용이하다.

이동에이전트를 이용한 다양한 네트워크관리 방법들이 연구되고 있다. LAN환경보다는 WAN환경이 이동 에이전트 방법이 더 좋은 응답시간을 보인다는 연구결과가 있고[11], 다중세그먼트 환경에서의 세그먼트 클러스터링 방식에 기반한 방법도 연구되었다[12]. 이중 네트워크에서의 이동에이전트를 이용한 네트워크 관리 방법도 연구되고 있고[13], Satoh[14]는 다양한 네트워크에 적용할 수 있는 재사용 가능한 이동에이전트에 대하여 연구하였다. Guo등은 자동으로 생성 가능한 plug-and-play 방식의 이동에이전트를 이용한 네트워크 관리방법을 제안하였다[15]. 에이전트 서버의 위치를 최적화 시키는 문제는 NP-hard 문제인데, Luo 등은 혼합형 유전 알고리즘을 이용한 방법을 제시하였다[16]. 네트워크관리의 신뢰성 향상을 위한 고장에 대비한 이동에이전트 기반의 네트워크 관리 방법도 연구되었다[17]. 한편, 이동에이전트를 이용할 경우 보안이 문제가 될 수 있음을 지적하고 이를 해결하기위한 방법도 제안되었다[18].

3. 각 네트워크 관리 모델의 통신량

다음은 중앙 집중형 관리와 이동 에이전트 기반의 네트워크 관리방식에 의한 통신량의 구체적인 비교를 수식을 통해 적용한 것이다. 먼저 네트워크 통신량의 수치화를 위한 가정으로 관리 대상의 수 N , 관리 대상에 보내질 질문 수 Q , 관리 대상에 보내질 데이터의 크기 d 로 놓을 때, 네트워크 패킷으로 바뀐 전체 데이터의 크기를 D 라 하면,

$$D = \alpha(d) + \beta(d)d$$

이다. 여기서 $\alpha(d)$ 는 통신 컨트롤을 위한 정보의 크기이고, $\beta(d)d$ 는 데이터 d 가 네트워크 패킷으로 바뀐 크

기이다. 위의 식을 데이터 정의할 수 있다.

$$D = \alpha(d) + \beta(d)d = \left\{ \frac{\alpha(d)}{d} + \beta(d) \right\} d$$

$$= \eta(d)d$$

여기서 $\eta(d) = \left\{ \frac{\alpha(d)}{d} + \beta(d) \right\}$ 이고

$\eta(d)$ 는 통신 오버헤드 함수를 의미하고

$\eta(d) > 1$ 을 항상 만족한다.

위 식을 통해 분석한 각 관리 모델의 통신량 계산은 다음과 같다[19].

RPC 기반의 중앙집중형 관리모델에서의 통신량, T_{RPC} 는 N 개의 관리대상 노드별로 Q 개의 질문 메시지를 보내야 하므로 다음식과 같이 정의된다.

$$T_{RPC} = \sum_{n=1}^N \sum_{q=1}^Q (\eta(I_q)I_q + \eta(R_{q,n})R_{q,n}) \quad (1)$$

(단, I_q 와 $R_{q,n}$ 은 각각 SNMP질문과 이에 따른 응답 결과의 크기)

이동 에이전트방식의 경우는 실행 코드와 응답결과가 통신량에 포함된다. 여기서 코드가 이동함에 따라 대신 별도의 질문메시지는 발생하지 않는다. 이동되는 코드는 노드를 거쳐 지나가도 크기가 일정하고 데이터 부분은 이동되는 노드가 증가함에 따라 선형적으로 증가하게 된다. 이동되는 코드의 크기를 C_{MA} 로 하고, 데이터의 크기를 $D_{MA,n}$ 라 할 때, 한 노드에서 발생되는 데이터의 크기는 다음과 같다.

$$D_{MA,n} = \sum_{m=1}^{n-1} \sum_{q=1}^Q R_{q,m} \quad (2)$$

(단, n=1인 경우, $D_{MA,n} = 0$ 이다. m은 관리대상 수이다)

따라서 전체 노드를 다 방문한 후의 총통신량 T_{MA} 는 다음과 같다.

$$T_{MA} = \sum_{n=1}^{N+1} (\eta(C_{MA} + D_{MA}) \times (C_{MA} + D_{MA})) \quad (3)$$

이를 토대로 할 때 중앙 집중형 방식의 관리는 관리

대상의 수 N과 관리 명령인 I_q 에 따라 전체 통신비용이 증가하게 되며, 이동 에이전트를 이용한 관리에서는 마찬가지로 이동해야 될 관리 대상의 수 N과 이동 코드와 데이터의 크기가 중요한 요소로 자리잡고 있다.

III. 분할관리 정책에 의한 네트워크 관리

1. 이동 에이전트 관리 모델의 개선

이동 에이전트를 이용한 네트워크 관리에서의 문제점은 관리대상의 수 N과 관리 명령 Q가 증가할수록 이동할 데이터의 크기, $D_{MA,n}$ 이 계속 증가한다는 것이다. 이에 대한 단점을 보완하기 위해 각각의 노드를 방문할 때 검색한 SNMP 변수들과 그 값은 누적시키지 않고 관리자에게 전송하여 주는 locker pattern을 적용시켜 좀 더 효율적인 관리를 해주는 모델도 제시되고 있다 [20]. 이 방식은 중앙집중형 RPC 방식에서의 통신횟수가 많아지는 단점과 이동 에이전트 방식의 경우 누적되는 데이터 양이 많아지는 단점을 일부 극복하기위한 방식으로서, 이동 에이전트가 방문한 노드에서 수집한 데이터를 누적하여 이동시키는 것이 아니라 방문한 노드의 임시 보관소에 데이터를 보관하고 다른 노드로 이동하는 방법을 사용한다. 임시 보관소의 데이터는 추후 RPC 방식으로 클라이언트로 전달된다. 그러나 이 방법은 별도의 보관을 위한 저장공간과 관리가 추가로 필요하다는 또 다른 문제점이 있다.

2. 분할 관리 정책

이동 에이전트를 이용한 네트워크 관리에서의 보다 효율적인 성능을 위해 다음을 제시한다. 먼저, 통신량과 관련된 가정은 II장과 같으며, 모든 노드간의 네트워크 대역폭이 동일한 균등 네트워크 환경이고, 전체 관리 대상 노드 N을 K개의 관리대상으로 분할한다고 가정했을 때, 분할영역은 $\lceil N/K \rceil$ 개의 영역으로 나누어진 다. 이렇게 일정한 영역으로 나누면 이동되는 응답 메시지의 누적 증가가 일정한 한도로 제한되는 효과가 있다. 기존의 이동 에이전트 방식의 네트워크 관리모델의 문제점은 관리할 노드 수가 증가할수록 이동되는 통신

량은 노드 수와 2차원적으로 비례하여 증가하게 된다. 따라서 노드 수가 일정 한도를 벗어나면 통신량이 급격하게 증가하는 문제가 발생하게 되는데, 이동 에이전트가 이동할 노드 수를 제한하면, 통신량이 급격히 증가하는 문제를 해결할 수 있다.

나누어진 영역의 수, $C = N/K$ 로 놓으면 (단, $N \bmod K = 0$), 임의의 한 분할영역, i 에서의 통신량, t_{MA,C_i} 는 다음과 같이 정의된다.

$$t_{MA,C_i} = \sum_{n=1}^{K+1} \eta(C_{MA} + D_{MA,n})(C_{MA} + D_{MA,n}) \quad (4)$$

단, $1 \leq i \leq \lceil N/K \rceil$, ($\lceil N/K \rceil$ 는 N/K 보다 작지않은 최소정수)

분할 관리에 의한 전체통신량, T_{DIV} 는 다음과 같다.

$$T_{DIV} = \sum_{i=1}^{\lceil N/K \rceil} t_{MA,C_i} \quad (5)$$

효율적인 통신을 위한 분할 조건은 T_{DIV} 가 T_{MA} 보다 작아야 한다.

분할된 관리영역 별로 각각의 이동 에이전트가 생성되고 파견되어 각각은 관리 대상의 수인 K 번의 순회 관리를 마치고, 복귀하게 된다. [그림 3]은 분할관리 정책에 의한 이동 에이전트 기반 네트워크 관리 방식을 나타내고 있다.

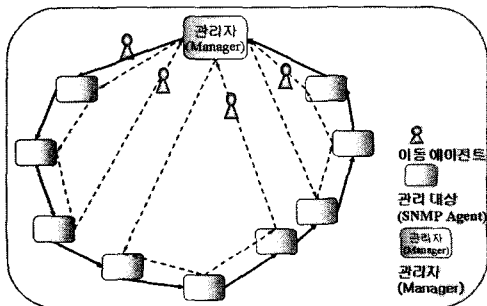


그림 3. 분할관리 정책에 의한 이동 에이전트 기반 네트워크 관리 방식

3. 통신 오버헤드

중앙 집중형 방식의 SNMP는 데이터 전송시 UDP방식을 사용한다. 따라서, $\eta(d) = \left\{ \frac{\alpha(d)}{d} + \beta(d) \right\}$ 에서의

$$\alpha(d) = 60 \text{ (IP + UDP + SNMP header),}$$

$\beta(d) = 1$ 로 놓을 수 있다. 오버헤드 파라미터는 각 관리 노드 N , SNMP 질문 수 Q 에 따른 6개의 MIB 값을 회수하도록 설정된 SNMP 변수의 크기는 72 bytes이다. 이 때, 오버헤드 함수 $\eta(I_q) = 1.83$ 이 된다. 마찬가지로 $\eta(R_{q,n}) = 1.83$ 이다. 이동 에이전트의 데이터 전송에는 TCP/IP가 사용되고 오버헤드 함수

$\eta(C_{MA} + D_{MA,n})$ 에서의 $\alpha(d)$ 는 TCP 연결 설정과 해제에 관한 통신량으로 200 bytes가 생성되며, $\beta(d)$ 는 TCP의 연결 메시지의 요구(request)와 승인(acknowledgment)에 추가되는 오버헤드로 다음과 같다[19].

$$\beta(d) = \frac{2H_{TCP/IP}}{d} \left[\frac{d}{P_{TCP}} \right] + 1 \quad (6)$$

이때, P_{TCP} 는 최대 TCP payload의 크기(1460 bytes)이며, $H_{TCP/IP}$ 는 TCP/IP 헤더의 크기이다. 또한, 생성되는 관리 에이전트의 클래스 크기

$(C_{MA} + D_{MA,n})$ 는 4420 bytes이다. 따라서, 오버헤드 함수 $\eta(C_{MA} + D_{MA,n}) = 1.09$ 가 된다.

그러나, 이동 에이전트에서는 관리 대상을 순회하면서, 질문 수 Q 에 따른 데이터 $D_{MA,n}$ 이 증가하게 되므로 오버헤드 함수의 크기도 그에 따라 변하게 된다.

4. 통신비용 계산

다음은 네트워크 통신비용을 알아보기 위해 전체 네트워크 통신량과 네트워크 소요시간과의 관계를 각 모델별로 수식을 통해 표현한 것이다. (단, 네트워크 지연 시간(delay)은 δ , 네트워크 대역폭(bandwidth)은 B 이다.)

$$C_{RPC} = 2 \sum_{n=1}^N \sum_{q=1}^Q \delta + \frac{T_{RPC}}{B} \quad (7)$$

$$C_{MA} = (N+1)\delta + \frac{T_{MA}}{B} \quad (8)$$

$$C_{MA,C_i} = (K+1)\delta + \frac{t_{MA,C_i}}{B},$$

$$C_{DIV} = \max\{C_{MA,C_i} | 1 \leq i \leq \lceil N/K \rceil\} \quad (9)$$

식 (7)은 RPC에 대한 전체 네트워크 소요시간에 대한 수식이다. 이는 N개의 관리 노드에 대한 Q번의 request와 response가 발생하므로 $2 \sum_{n=1}^N \sum_{q=1}^Q \delta$ 의 네트워크 지연시간이 추가로 소요되며, (8)은 이동 에이전트에 대한 소요시간으로 이동에 소요되는 $(N+1)\delta$ 의 지연율이 필요하다. (9)도 각 분할 영역에 대한 지연율 $(K+1)\delta$ 가 발생하며, 분할 영역에서의 총 소요시간 C_{DIV} 는 각각의 영역 별 시간, C_{MA,C_i} 중에서 최대소요시간을 필요로 하는 관리 에이전트의 소요시간이다.

IV. 실험 결과 및 고찰

실험을 위한 이동 에이전트 플랫폼으로는 자바를 기반으로 한 IKV++의 Grasshopper[21]를 이용하였고, 각각의 관리대상이 되는 SNMP노드는 팬티엄 PC 10대를 이용하였다. 각 컴퓨터는 10Mbps의 대역폭을 지원하는 이더넷 LAN으로 하나의 세그먼트내에 연결되어 있으며, 또한 에이전트를 관리하는 네트워크 관리자 컴퓨터도 동일한 세그먼트내에 배치되었다.

10대의 SNMP 노드를 통해 관리대상의 수를 증가시키기 위해서는 일반적인 이동 에이전트를 이용하는 방법에서는 관리 에이전트가 관리대상 노드를 순회하는 수에 따라 각각 10회씩 증가하도록 하였다. 한편, 분할 관리를 위한 이동에이전트의 이주는 관리자 컴퓨터에서 분할된 관리 영역의 수에 따라 에이전트를 생성하여 파견하면 관리 에이전트는 각각의 노드를 순회하여 관리하게 된다. 예를들면, 100개의 관리 대상의 실험을 위해서는 관리 에이전트는 10개가 생성되며 각 에이전트는 관리자에서 파견되어 각각의 노드를 한번씩 순회하게 된다. 이 경우 각 에이전트의 1회 순회가 마치게 되

면 100개의 관리대상을 관리할 수 있도록 가정하였다. 중앙집중형 방식의 SNMP 모델로는 AdventNet tkdml SNMP 패키지[22]를 사용하였다.

표 1. IP층의 효율도 계산에 필요한 관리 객체

Variable Name (Managed Object)	OID (Object Identity)
IpInReceives	1.3.6.1.2.1.4.3
IpInHdrErrors	1.3.6.1.2.1.4.4
IpInAddrErrors	1.3.6.1.2.1.4.5
IpInUnknownProtos	1.3.6.1.2.1.4.7
IpInDiscards	1.3.6.1.2.1.4.8
IpOutRequests	1.3.6.1.2.1.4.10
IpOutDiscards	1.3.6.1.2.1.4.11
IpOutNoRoutes	1.3.6.1.2.1.4.12
IpReasmFails	1.3.6.1.2.1.4.16
IpFragFails	1.3.6.1.2.1.4.18
IpRoutingDiscards	1.3.6.1.2.1.4.23

만일 이동 에이전트가 수행해야 될 네트워크 관리 동작이 각 관리 대상에 대한 IP 층의 효율도(Utilization)를 계산해야 된다면, 이동 에이전트는 [표 1]과 같은 내용을 SNMP Agent에게 요구를 하고 SNMP Agent는 해당 MO(OID로 표현되며 MIB안에 있음)의 값을 전달하게 된다. 관리자는 이동 에이전트로부터 얻어진 정보를 통하여 순간 IP Input Packet량, 순간 IP Output Packet량, 순간 IP Error량의 평균을 산출하고 이를 토대로 IP층의 효율도(Utilization)를 아래 식 (10)을 통해 IP층의 효율도를 계산 할 수 있게 된다.

$$\left[\frac{\text{순간 } IPErrorPacket \text{ 량의 평균}}{\text{순간 } IPInputPacket \text{ 량의 평균} + \text{순간 } IPOutputPacket \text{ 량의 평균}} \right] \times 100(\%) \quad (10)$$

그러나, 본 실험에서 사용한 OID(Object Identity)는 본 논문의 특성상 특정 관리 작업에 큰 의미를 부여하지 않아도 되기 때문에 특정 관리 작업에 관련된 값을 정하지 않고, 아래 [표 2]의 코드 부분에서 볼 수 있듯이 “1.3.6.1.2.1.1”, “1.3.6.1.2.1.2”, “1.3.6.1.2.1.4”, “1.3.6.1.2.1.6”, “1.3.6.1.2.1.7”, “1.3.6.1.2.1.11”와 같은 임의의 값을 설정하도록 하였다.

표 2. 이동 에이전트가 관리할 질문의 예제

```
public class query extends MobileAgent
{
    boolean terminate;
    String oids[] =
    {
        "1.3.6.1.2.1.1",
        "1.3.6.1.2.1.2",
        "1.3.6.1.2.1.4",
        "1.3.6.1.2.1.6",
        "1.3.6.1.2.1.7",
        "1.3.6.1.2.1.11"
    };
    :
}
```

다음은 식 (7)과 (8), (9)에 대한 전체 소요시간을 실험을 통해서 입증하고, 성능 평가를 보인 그래프이다.

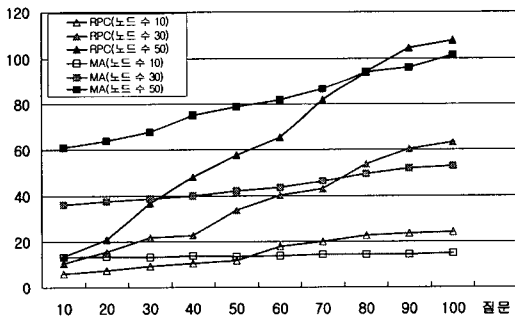


그림 4. 중앙 집중형(RPC)방식과 이동 에이전트 관리방식의 통신비용 비교

[그림 4]는 중앙집중형 RPC 방식과 이동 에이전트 관리방식에 대하여 각각 질문 수 Q와 관리 대상 수 N을 증가 시키면서 실험한 결과를 노드 수 별로 표시한 것이다. 그래프의 가로축은 질문 수를 표시하고 세로축은 통신비용을 초 단위로 표시하고 있다. 이 실험의 결과에 의하면 중앙집중형 RPC방식에 비해 이동 에이전트 방식이 성능의 커다란 향상이 없음을 알 수 있다.

질문 수가 적을 경우에는 중앙집중형 RPC 방식이 이동 에이전트 방식에 비하여 통신량이 오히려 작은 것을

알 수 있다. 이는 질문 수가 적은 경우 이동 에이전트의 통신 오버헤드가 상대적으로 크기 때문이다. 질문 수가 많아짐에 따라 이동 에이전트 방식이 중앙집중형 RPC 방식에 비하여 상대적 통신량이 약간 작아지는 것을 알 수 있다. 특히 노드 수가 많은 경우 질문 수가 많아지더라도 중앙집중형 RPC 방식에 대한 이동 에이전트 방식의 성능향상은 큰 차이가 없다. 관리대상 노드 수가 10개 정도일 때 질문 수의 증가에 따른 통신량의 증가율의 기울기가 제일 완만하다. 이 자료를 근거로 하여 분할 영역관리 대상수를 10으로 설정하였다. 즉 K=10인 경우, 노드 수가 50개와 100개에 대한 분할 관리영역수, C=5, 10으로 각각 나누어진다. 이동 에이전트는 C의 수만큼 생성하고 각각 정해진 영역 범위 내에서 이동하게 하였다. 이에 대한 실험 결과를 [그림 5]와 [표 3]에 각각 제시하였다.

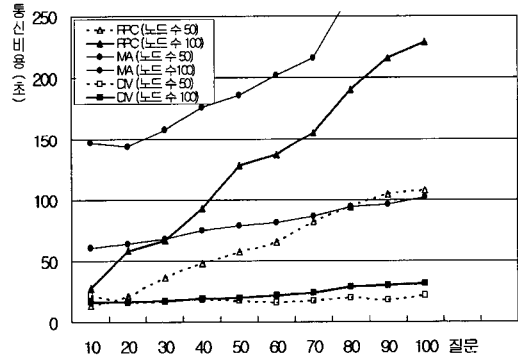


그림 5. 중앙 집중형(RPC)방식과 분할 관리 정책 방식의 통신비용 비교

표 3. 노드 수와 질문 수에 따른 통신비용(초)

노드 질문	50			100		
	RPC	MA	DIV	RPC	MA	DIV
10	13.4	60.8	21.6	27.6	146.7	16.3
20	21	63.9	15.7	58	143.5	16.6
30	36.6	67.7	16.6	66.3	158	17.4
40	48	75.2	18.7	93.2	176.5	19.2
50	57.6	78.7	17.2	128	185.6	19.9
60	65.2	81.7	16.1	137.2	201.9	21.8
70	81.7	86.6	17.5	155.9	216	23.6
80	94.2	94.2	19.5	190.2	267.6	28.8
90	104.4	96.2	18.1	216.2	273.7	29.9
100	107.8	101.6	21.6	228.7	276.2	31.3

여기서는 중앙 집중형 RPC 방식과 이동에이전트 방식 그리고 제한한 분할관리 정책과의 비교한 결과를 보여주고 있는데 중앙 집중형 방식은 질문 수에 비례하여 증가하며, 또한 노드의 수에 비례하여 증가하는 것을 확인할 수 있고 이동 에이전트 방식은 노드의 수가 적은 경우에는 완만한 기울기를 보이다가 노드의 수가 증가할수록 통신환경과 컴퓨팅 능력 및 질문 메시지의 크기 등의 영향에 따라 급격한 증가를 보이게 된다. 그러나 제한한 분할 관리 정책에서는 질문의 수와 노드의 수에 크게 영향을 받지 않고 통신량이 완만하게 증가함을 알 수 있다.

분할관리정책에 의한 상대적인 통신비용 효율 E_{DIV} 를 중앙집중형 RPC 방식에 의한 통신비용 C_{RPC} 와 분할관리정책에 의한 통신비용 C_{DIV} 와의 비율이라고 정의하고,

$$\text{즉 } E_{DIV} = \frac{C_{RPC}}{C_{DIV}}$$

기존의 이동 에이전트 관리방식에 의한 상대적인 통신비용 효율 E_{MA} 를 중앙집중형 RPC 방식에 의한 통신비용 C_{RPC} 와 이동 에이전트 관리방식에 의한 통신비용 C_{MA} 와의 비율이라고 정의할 때,

$$\text{즉 } E_{MA} = \frac{C_{RPC}}{C_{MA}}$$

노드의 수와 질문 수가 증가함에 따라 E_{DIV} 는 1보다 큰 값으로 증가함을 알 수 있다. [표 4]에 노드 수 50과 100에 대한 질문 수 10개부터 100개까지의 각각의

경우에 대해서 상대적인 통신비용 효율 E_{DIV} 와 E_{MA} 의 측정값을 나타내었다. 예를 들면 노드의 수가 50개이고 질문 수가 10일 때는 $E_{DIV} = 13.4/21.6 = 0.62$ 이고, $E_{MA} = 13.4/60.8 = 0.22$ 로서 기존의 이동에이전트 관리방식에 비하여 분할정책에 의한 관리방식의 상대적 통신비용 효율이 약 3배정도 우수하고 중앙집중형 RPC 방식에 비하여 효율이 나쁘게 측정되었다. 그러나 질문 수가 20인 경우에는, $E_{DIV} = 1.34$ 와 $E_{MA} = 0.33$ 으로서 중앙집중형 RPC 방식에 비해서도 효율이 34% 정도 더 우수할 뿐만 아니라 기존의 이동 에이전트 관리방식에 비하여 효율이 4배정도 더 우수함을 알 수 있다. 질문 수가 100인 경우에는 $E_{DIV} = 4.99$ 와 $E_{MA} = 1.06$ 으로서 중앙집중형 RPC 방식에 비하여 분할관리 정책에 의한 방식이 5배정도 더 우수하고 기존의 에이전트 방식에 비하여 4.7배 정도 더 우수함을 알 수 있다. 노드 수가 100개이고 질문 수가 100개인 경우는 $E_{DIV} = 7.31$ 와 $E_{MA} = 0.82$ 로서 제안된 분할관리 정책에 의한 방식이 앞에서 비교한 경우들 보다 상대적으로 더 우수한 결과를 보이고 있다.

V. 결론 및 향후 연구과제

본 논문에서는 네트워크 관리 시스템에서의 중앙 집중형 방식의 문제점을 해결하기 위해 이동 에이전트를 이용한 관리 방식에서의 보다 효율적인 성능향상을 위한 방안을 제시하였다. 이동 에이전트를 이용한 네트워크 관리에 있어서의 네트워크 통신량은 이동하는 에이전트의 크기에 따라 변하는데 본 논문에서는 이에 영향을 주는 파라미터인 관리에 필요한 질문 수와 관리 대상의 수에 따른 네트워크의 전체 통신비용의 증가를 알아보고 이를 줄이기 위한 분할 관리 정책을 제시하고 성능평가 및 비교 분석을 통해 제시한 정책의 효율성을 보였다.

실험은 동일한 세그먼트내의 노드들로 구성하여 실험을 하였는데, LAN의 특성상 노드사이의 에이전트 이동이 동시에 이루어질 수 없기 때문에 다중 세그먼트에 분산된 노드 환경에 비하여 병렬성이나 동시성에 대한

표 4. 노드 수와 질문 수에 따른 상대적인 통신비용 효율

노드 질문	$E_{DIV}(50)$	$E_{MA}(50)$	$E_{DIV}(100)$	$E_{MA}(100)$
10	0.22	0.62	0.19	1.69
20	0.33	1.34	0.40	3.49
30	0.54	2.20	0.42	3.81
40	0.64	2.57	0.53	4.85
50	0.73	3.35	0.69	6.43
60	0.80	4.05	0.68	6.29
70	0.94	4.67	0.72	6.61
80	1.00	4.83	0.71	6.60
90	1.09	5.77	0.79	7.23
100	1.06	4.99	0.83	7.31

이득이 없다. 따라서 다중 세그먼트에서 이동 에이전트를 분산시켜 실험하는 것 보다 최악의 조건에서 실험하였기에, 제시한 방법을 다중 세그먼트 환경에 적용하여 실험한다면 더 나은 결과가 나올 것으로 기대된다.

향후 연구과제로는 네트워크의 규모에 따라 최적의 관리영역의 크기, 즉 생성될 이동 에이전트의 수를 결정하는 방법이다. 또한 단일 세그먼트 환경에서만이 아닌 다중 세그먼트에 관리대상 노드를 배치한 좀 더 큰 규모에서의 네트워크 관리에 대한 다양한 환경에서의 실험이 필요하며, 네트워크의 각 노드의 부하와 통신비용이 불균일한 일반적인 경우에 대한 영역분할 방법 등의 연구도 고려되어야 할 부분이다. 통신비용외에 이동 에이전트 시스템관리 비용을 고려한 연구도 필요하다고 사려된다.

참고 문헌

- [1] J. Case, M. Schoffstall, and J. Davin, *A Simple Network Management Protocol (SNMP)*, RFC 1157.
- [2] W. Stallings, *SNMPv2 and RMON*, Addison Wesley, SNMP, 1996.
- [3] A. Bieszczad, T. White, and B. Pagurek, "Mobile Agents for Network Management," *IEEE Communications Surveys*, Sep. 1998.
- [4] D. Gavalas, D. Greenwood, M. Ghanbari, and M. O'Mahony, "An Infrastructure for Distributed and Dynamic Network Management based on Mobile Agent Technology," *IEEE International Conference on Communications (ICC'99)*, 1999.
- [5] A. Sahai and C. Morrin, "Towards Distributed and Dynamic Network Management," *Proc. of the IEEE/IFIP Network Operation and Management Symposium(NOMS)*, New Orleans, Louisiana, USA, Feb. 1998.
- [6] G. Susilo, A. Bieszczad, and B. Pagurek, "Infrastructure for Advanced Network Management Based on Mobile Code," *Proc. of the IEEE/IFIP Network Operation and Management Symposium(NOMS)*, New Orleans, Louisiana, USA, Feb. 1998.
- [7] G. Goldszmidt and Y. Yemini, "Distributed Management by Delegation," *In Proceedings of the 15th International Conference on Distributed Systems*, June 1995.
- [8] G. Goldszmidt, "Distributed System Management via Elastic Servers," *In Proceedings of the IEEE First International Workshop on Systems Management*, April 1993.
- [9] 박종태, "망관리 기술 및 연구동향", 한국통신학회 통신망운용관리연구회, KNOM Review, Vol.1, No.1, pp.1-5, 1998(2).
- [10] D. Chess, C. Harrison, and A. Kershenbaum, *Mobile Agents: Are They a Good Idea?*, 1995.
- [11] 이정우, 최재혁, 이우석, 최상방, "네트워크 관리 시스템의 해석적 모델 및 성능 평가", 전자공학회 논문지, 제41권 TC편, 제9호, pp.17-28, 2004(9).
- [12] 노정민, 윤종호, "세그먼트 클러스터링 방식을 이용한 이동 에이전트형 망 관리 시스템의 구현 및 성능 분석", 한국통신학회논문지, 제26권, 제2호, pp.187-197, 2001(2).
- [13] J. Lefebvre, S. Chamberland, and S. Pierre, "A Network Management Framework using Mobile Agent," *IEEE CCECE 2003 Canadian Conference on Electrical and Computer Engineering Vol.2*, pp.737-740, May 2003.
- [14] I. Satoh, "Building Reusable Mobile Agents for Network Management," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, Vol.33, pp.350-357, Aug. 2003.
- [15] N. Guo, T. Gao, and H. Zhao, "Distributed Plug-and-Play Network Management Model Based on Mobile Agents," *2004 IEEE International Conference on e-Technology, e-Commerce and e-Service*, pp.487-491, March

2004.

[16] X. Luo, P. Yan, C. Guo, and Y. Tang, "Optimal placement and deployment strategies in mobile agent-based network management," IEEE 2002 International Conference on Communications, Circuits and Systems, Vol.1, pp.753-757, June/July 2002.

[17] J. H. Li and G. H. Xu, "A New Network Management Framework Design and Application Realization," Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies(PDCAT'05), pp.890-894, Dec. 2005.

[18] K. Alexandros and S. Joseph, "A Trustworthy Mobile Agent Infrastructure for Network Management," 10th IFIP/IEEE International Symposium on Integrated Network Management, pp.383-390, May 2007.

[19] G. Picco and M. Baldi, "Evaluating the Tradeoffs of Mobile Code Design Paradigms in Network Management Applications," In Proceedings of the 20th International Conference on Software Engineering (ICSE'98), Kyoto (Japan), April 1998.

[20] 권혁찬, 김홍환, 유관중, "네트워크 관리를 위한 이동 에이전트의 성능평가", 한국정보처리학회 논문지, 제8-C권, 제1호, pp.68-74, 2001.

[21] <http://www.ikv.de/products/grasshopper>

[22] <http://www.adventnet.com/products>

저 자 소 개

이 재 형(Jae-Hyoung Lee)

정회원



- 2000년 2월 : 금오공과대학교 컴퓨터공학부(공학사)
- 2002년 2월 : 금오공과대학교 컴퓨터공학과(공학석사)
- 2003년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과 박사과정

<관심분야> : 무선 네트워크 및 보안, 차세대 통신망

오 길 호(Gil-Ho Oh)

정회원



- 1980년 2월 : 서울대학교 전기공학부(공학사)
- 1980년 2월 : 서울대학교 컴퓨터공학과(공학석사)
- 1992년 12월 : University of Florida, Computer and Information Science Engineering Dept.(공학박사)
- 1983년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학부 교수

<관심분야> : 객체지향 시스템, 병렬/분산처리 소프트웨어공학, 임베디드 시스템