

네트워크 관리 시스템의 성능 향상을 위한 연구

양진석*, 김현주**, 장범환**, 정태명**

*성균관대학교 정보공학과

**성균관대학교 전기전자 컴퓨터 공학부

e-mail: {jsyang, hjkim, bhchang}@rtlab.skku.ac.kr ,
tmchung@ece.skku.ac.kr

The Research for the Efficiency Improvement of Network Management System

Jin-Seok Yang*, Hyun-Joo Kim**, Bum-Hwan Chang**, Tai M. Chung**

*Dept of Information Engineering, Sungkyunkwan University

**Dept of Electric & Computer Engineering, Sungkyunkwan University

요약

오늘날 네트워크는 급격한 확장으로 인하여 복잡성을 더해가고 있으며, 그에 따라 네트워크 관리를 위한 정보의 양도 계속 늘어나고 있다. 네트워크 확장에 따른 사용자 데이터뿐만 아니라 관리 데이터의 증가로 인하여 병목 현상, 폭주등 네트워크 문제를 가속화시킴으로 효율적인 관리 정보의 처리에 대한 필요성 점차 높아가고 있다. 본 논문에서는 이를 해결하기 위한 방안으로 관리 정보의 증가를 효율적으로 처리하는 방법과 네트워크 관리 시스템에 적용하는 방법을 기술한다.

1. 서론

오늘날의 네트워크는 네트워크 장비, 이종의 시스템, 사용자의 급격한 증가로 인해 점차 확장됨에 따라 이러한 환경을 관리하는데 있어 많은 어려움을 겪고 있으며, 네트워크 관리를 위한 정보도 급격히 늘어나고 있어 네트워크 관리를 위한 정보를 효율적으로 처리해야할 필요성이 대두되었다.

현재 네트워크 관리 시스템은 관리 노드까지의 거리나 관리를 위한 작업 수행 시간에 상관없이 중앙 집중적인 구조를 하고 있다. 이러한 구조의 네트워크 관리 시스템은 꾸준히 증가하는 노드들을 하나의 네트워크 관리 시스템이 관리하고 있으므로 시스템의 과부하와 관리 데이터의 집중으로 네트워크 내의 트래픽도 증가하게 되어 병목 현상등을 야기한다.

본 논문에서는 이러한 중앙 집중식 구조의 단점을 보완하고 최소의 관리 데이터로 최대의 효과를 얻기 위해서 네트워크 관리 시스템의 분산처리와 네트워크 관리에 필요한 정보들을 효율적으로 가져오기 위한 방법에 대해 기술한다.

본 논문의 구성은 2장에서 관련 연구에 대한 내용들을 언

급하고 3장에서는 성능 향상을 위한 연구, 4장에서는 결론 및 향후 연구 과제를 다룬다.

2. 관련 연구

2.1 MIB

MIB(Management Information Base)이란 네트워크 상의 시스템들이 모니터링되고 구성되며 제어될 수 있기 위해 필요한 관리 정보들을 정의하고 있는 명세서이다.

MIB을 구성하는 모든 관리 객체는 계층적으로 구성되고, MIB이 이루는 트리는 말단 노드의 객체만이 실제 값을 가지며, 각각의 객체는 객체 식별자(Object Identifier)를 가진다. 객체 식별자는 MIB내에서 관리 객체를 유일하게 지정하는 식별자이다[3].

2.2 분산 처리

분산이란 컴퓨터 프로그램이나 데이터가 네트워크 상에, 즉 대개 두 대 이상의 컴퓨터에 걸쳐 있을 때, 이것을 분산컴퓨팅 이라고 부르며 두 대 이상의 컴퓨터가 같은 작업을 수행하는 것을 분산 처리라고 한다. 분산 처리의 장점을 5가지로 나누어 보면 다음과 같다.

- 1) 보안과 캡슐화

시스템 설계자는 사용자와의 상호작용을 제한할 수 있다. 즉, 시스템 설계자는 사용자가 시스템 전체를 접근하지 못하게 설계하여 시스템 보안과 캡슐화를 할 수 있다.

2) 분산 데이터베이스

전체 데이터를 저장하기 위해 여러 시스템이 데이터 저장소를 가질 수가 있다.

3) 문제의 빠른 해결

여러 대의 컴퓨터가 어떤 문제에 대해 동시에 작업을 수행하므로 한 대의 컴퓨터가 처리하는 것 보다 훨씬 빠르다.

4) Redundancy를 통한 보안

여러 대의 컴퓨터가 같은 일을 처리할 때는 redundancy를 통해 안정성을 제공할 수 있다.

5) 동시 처리

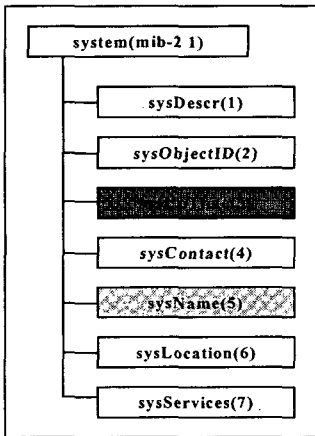
여러 대의 컴퓨터와 여러 명의 사용자가 동시에 일을 할 수 있다.

즉, 분산 처리는 각각의 시스템이 데이터 저장소와 처리기들을 가지며 여러 처리기들이 동시에 여러 작업을 수행함으로써 성능 향상과 데이터 복사본을 여러 곳에 유지함으로써 신뢰도의 증가, 네트워크에 새로운 처리기 등을 추가함으로써 시스템의 확장성을 제공한다[1].

3. 성능 향상을 위한 연구

3.1 MIB 관련 연구

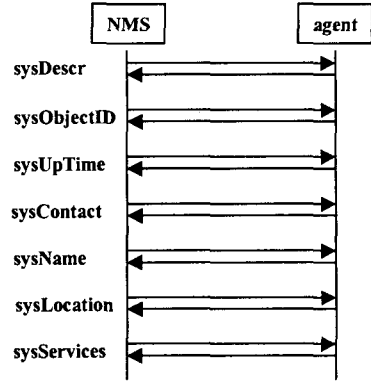
본 장에서는 네트워크 관리 시스템의 성능 향상과 효율적인 관리 정보 처리를 위해 MIB객체에 대해 정적 정보와 동적 + 정적 정보 그리고 동적 정보로 나누어 보았다.



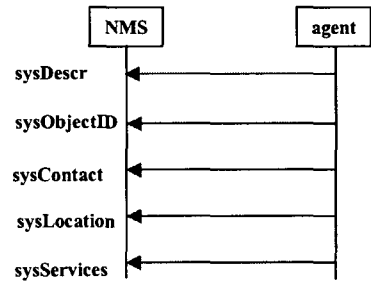
[그림 1] system 그룹

system 그룹에서 흰색 바탕의 객체들은 거의 바뀌지 않는 정보 즉, 정적 정보를 나타내고, 회색 바탕은 정적 + 동적 정보를, 검은 바탕의 객체는 동적 정보를 나타낸다.

정적 정보는 매번 폴링해서 가져오는 것보다 정적 정보들이 바뀌는지의 여부를 모니터링하고 있으면서 값이 바뀌면 그 값을 관리 시스템으로 알려준다.



[그림 2] 기존 시스템의 통신 방식



[그림 3] 새로운 통신 방식

기존의 관리 시스템은 [그림 2]에서 보는 바와 마찬가지로 모든 정보에 대해서 요구하고 그에 대한 응답을 얻는다. 그러나, 정적인 정보에 대해서는 계속 정보를 요구할 필요가 없다. [그림 3]에서와 같이 정보가 바뀌었을 때 관리 시스템으로 변화된 값을 알리는 것이 효율적이며 동적인 정보는 그 값이 수시로 바뀌므로 관리 시스템이 요구할 때마다 가져오는 것이 효율적이다. 동적 + 정적인 정보는 동적 정보보다 자주 바뀌지는 않지만, 정적 정보보다는 자주 바뀌는 값들이다. 이러한 값들이 네트워크 관리 정책에 따라 틀려진다. 관리 시스템이 자주 요구하는 값이면 폴링에 의해서 값을 가져올 수도 있고, 그렇지 않으면 주기를 두어 트랩으로 시스템에 알릴 수도 있다.

예를 들어 특정 컴퓨터의 system 그룹은 [그림 4]와 같은 관리 정보를 갖고 있다.

네트워크 관리 시스템이 system 그룹에 대한 관리 정보를 요구하면 노드가 다운되지 않았을 경우 [그림 4]에서 밑줄 친 정보를 응답한다. 그 중에서 3번과 5번을 제외한 1, 2,

4, 6, 7번 값들 즉, 정적 정보들은 자주 변하는 값이 아니기 때문에 매번 폴링할 필요가 없다.

```

1: sysDescr.0
Sun SNMP Agent.SUNW.Ultra-Enterprise
2: sysObjectID.0
enterprise.42.2.1.1
3: sysUpTime.0
0 days 07h:58m:49s.18th(2872918)
4: sysContact.0
System administrator
5: sysName.0
ece
6: sysLocation.0
System administrators office
7: sysServices.0
72
    
```

[그림 4] system 그룹의 폴링 결과

네트워크 관리 시스템은 이미 이전의 정적 정보 값들을 저장되어 있는 데이터 베이스를 갖고 있다. 따라서, [그림 3]과 같이 바뀐 객체의 값만 트랩으로 받아 데이터 베이스를 변경하면 그에 따른 오퍼레이션과 네트워크내의 관리 트래픽은 상당히 줄어들게 된다.

위에서는 system 그룹에서만 예를 든 것이고, 다른 MIB 그룹에서도 위와 같이 정적 정보와 동적 정보 그리고 정적 + 동적 정보를 정의하고 관리 시스템이 요구를 할 때 마다 폴링하는 것이 아니라 객체의 값이 바뀌었는지의 여부를 먼저 알아낸 후 폴링하는 것이 네트워크내의 부하를 줄일 수 있고, 시스템이 처리해야 하는 작업도 적어질 수 있다.

여기서 중요한 점이 있다. 값의 변화 빈도에 따라 MIB 객체를 3가지로 그룹화했는데, 이러한 그룹화는 상당히 중요하다. 그룹화를 어떻게 하느냐에 따라 네트워크나 관리 시스템의 성능을 좌우하기 때문이다.

3.2 에이전트 관련 연구

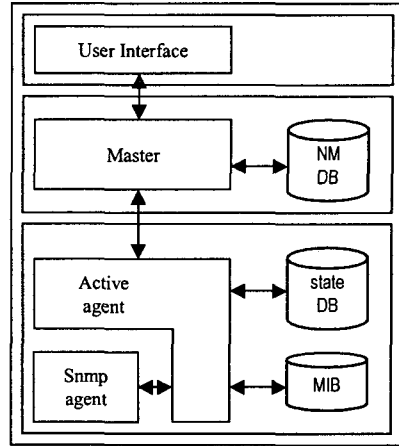
본 장에서는 SNMP 에이전트와 Active 에이전트에 대해서 기술한다.

현재 서버급 컴퓨터와 모든 네트워크 장비는 SNMP 에이전트가 탑재된 상태에서 출시되고 있다.

SNMP 에이전트는 트랩 메시지를 제외한 모든 메시지에 대해 폴링에 대해서만 응답하는 수동적 에이전트(passive agent)이다. 트랩 메시지는 관리 시스템의 요구없이 관리 대상에서 발생한 사건을 에이전트가 자체적으로 관리 시스템으로 보내는 메시지이다.

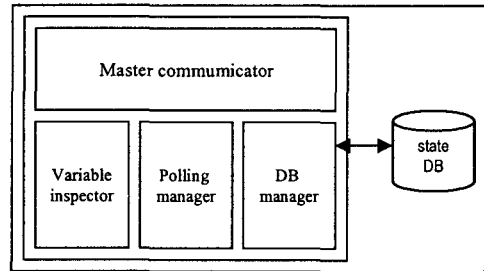
지금부터 관리 정보에 대한 효율적인 처리를 위해 Active 에이전트(A-agent)를 설명한다.

A-agent는 정적 정보가 바뀌었는지의 여부를 저장하는 DB(state DB)를 갖고 있다. A-agent는 SNMP 에이전트의 동작을 모니터링하고 있으며, MIB의 어떠한 객체가 바뀌면 그 객체의 상태를 true로 state DB에 저장한다.



[그림 5] Active 에이전트의 처리 구조

Master로부터 관리 정보 요청을 받은 A-agent는 먼저 state DB를 접근하여 정보가 바뀌었는지를 검사한다. 만약 바뀌었으면 바뀐 값만을 Master로 보내고 state DB 값을 false로 바꾼다. Master로 보내어진 데이터는 Master DB에 업데이트된다.



[그림 6] Active 에이전트 구조

[그림 6]은 A-agent의 구조를 나타낸다.

Master Communication 모듈은 Master와의 통신을 위한 모듈이다.

Variable inspector 모듈은 Master로부터 요청이 왔을 때 요청한 관리 정보의 state DB를 보고 바뀌었는지의 여부를 검사하는 모듈이다.

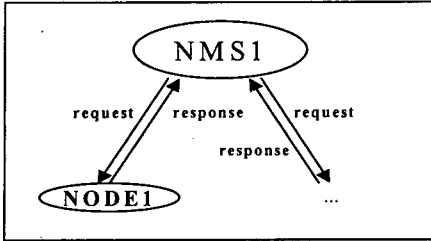
Polling Manager는 SNMP 에이전트의 폴링을 담당하는 모듈이다.

DB manager는 DBMS(database management system)와

통신하고 결과를 가져오기 위한 모든 처리를 한다.

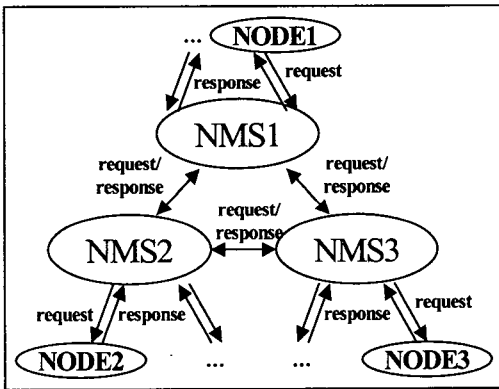
3.3 네트워크 관리 시스템에 적용

본 장에서는 2.2절에서 기술한 분산처리와 3.2절에서 기술한 에이전트를 분산 관리 체제에 적용하여 보고자 한다.



[그림 7] 중앙 집중식 NMS

[그림 7]은 중앙 집중식 네트워크 관리 시스템을 나타내고



[그림 8] 분산 처리 기반의 NMS

[그림 8]은 분산처리기반의 네트워크 관리 시스템을 나타낸다. [그림 7]의 NMS1은 관리 노드 전체가 하나의 네트워크 관리 시스템에 저장되어 있지만, [그림 8]의 NMS1과 NMS2 그리고 NMS3은 각각 독립된 데이터 베이스를 가지고 있고, 각 시스템의 데이터 베이스에는 시스템의 관리 노드와 이에 대한 여러 가지 관리 정보가 저장되어 있다. 각각의 NMS는 계층 구조가 아닌 동등한 계층의 시스템이다.

네트워크 관리 시스템 관리자는 폴링 interval을 두어 등록된 노드의 관리 정보를 주기적으로 요청하고 SNMP 에이전트가 인스톨된 노드는 네트워크 관리 시스템의 요청에 응답한다.

[그림 8]의 NMS1, NMS2, NMS3는 각각의 등록된 관리 노드에 폴링하고 에이전트가 응답한 데이터를 수집하여 각각의 데이터 베이스에 저장한다.

만약, 네트워크 관리 시스템이 등록되지 않은 노드의 정보를 요구할 경우 즉, NMS1이 NMS2에 등록된 노드의 정보를 요구할 경우는 NMS1은 NMS2에 요청하고 NMS2는 자신의 데이터 베이스에서 정보를 가져와 응답한다. 이와 같은 시나리오가 이루어지기 위해서는 네트워크 관리 시스템들 사이에 이벤트를 처리할 수 있도록 프로토콜이 정의되어 있어야 하고, 메시지를 처리할 수 있는 처리기가 있어야 한다.

[그림 7]와 [그림 8]의 관리 노드의 개수가 서로 같다고 가정하면, 그 처리는 직관적으로도 [그림 8]의 처리가 빠름을 알 수 있다.

여기서, NMS1은 NMS2를 유일하게 구별할 수 있는 index를 가져야 한다. 그리고, NMS1이 관리 노드를 추가 시키고자 할 때는 추가시킬 노드가 NMS2 혹은 NMS3등의 데이터베이스에 존재하는지의 여부를 검사한 후 없을 경우에만 추가할 수 있도록 해야 한다.

4. 결론 및 향후 연구 과제

본 연구에서는 기존의 중앙 집중식 네트워크 관리의 단점을 보완하고자 분산처리를 통해서 많은 양의 관리 정보를 효율적으로 처리할 수 있는 방법을 기술하였다. 관리 정보의 효율적인 처리를 위해서 system MIB을 예로 MIB 객체의 그룹화를 제안하였고, Active 에이전트의 개념도 기술하였다. 그리고, 실제 네트워크 관리 시스템에 적용하여 보았다.

향후 연구 과제로는 또 다른 standard MIB 그룹의 그룹화, 에이전트의 기능 확장, 네트워크 관리 시스템에서 실시간 정보를 효율적으로 처리하기 위한 연구가 필요하다.

참고문헌

[1]Behrouz A. Forouzan, "Data Communications and Networking", 2nd Ed. McGraw Hill.
 [2]M. Rose & K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", RFC1155.
 [3]M. Rose & McCloghrie, "Concise MIB Definitions", RFC1212.
 [4]William Stallings, "SNMP, SNMPv2, SNMPv3, and RMON1 and 2" 3rd Ed. Addison Wesley.