논문 2004-41TC-9-3

네트워크 관리 시스템의 해석적 모델 및 성능 평가

(Analytical Models and Performance Evaluations of Network Management Systems)

이 정 우*, 최 재 혁*, 이 우 석*, 최 상 방**

(Jung Woo Lee, Jae Hyuk Choi, Woo Seok Lee, and Sang Bang Choi)

요 약

인터넷이 급속히 발전하여 통신 하부구조가 됨에 따라 네트워크의 많은 구성요소를 체계적으로 관리하는 네트워크 관리 시스템은 인터넷의 필수 요소가 되고 있다. 네트워크 규모의 급속한 성장은 기존의 SNMP(Simple Network Management Protocol), CMIP(Common Management Information Protocol) 등을 기반으로 한 클라이언트-서버(client-server) 관리 패러다임으로는 한계를 가진다. 따라서 네트워크를 효율적으로 관리하기 위해서 분산형(distributed) 패러다임인 이동에이전트(Mobile Agent)를 네트워크 관리에 이용하려는 연구가 최근에 많이 이루어지고 있다. 본 논문에서는 중앙 집중형의 SNMP, 분산형의 이동에이전트, 그리고 이들 두 접근 방법의 단점을 극복하기 위한 이동에이전트의 한 형태인 혼합모드의 해석적 모델을 제안하고 그 성능을 비교 분석한다. 제안한 해석적 모델을 네트워크 응답 시간에 중점을 두어 성능 평가 한 결과 LAN에서는 대체적으로 SNMP가 유리한 반면 WAN에서는 네트워크 환경에 따라 이동에이전트 또는 혼합모드가 더 좋은 응답 시간을 보임을 알 수 있다.

Abstract

Since the Internet has grown into a large communication infrastructure, the network management system to systematically manage the large number of network components has become an essential part of the Internet. The rapid growth of network size has brought into question the scalability of the existing centralized model, such as SNMP (Simple Network Management Protocol) and CMIP (Common Management Information Protocol). Thus, for efficient network management, researches about mobile agent have also been performed recently. This paper presents analytical models of centralized approach based on SNMP protocol, distributed approach based on mobile agent, and mixed mode to make up for shortcomings of SNMP and mobile agent. We compare the performance of these analytical models based on network management response time. Experiment results show that performance of mobile agent and the mixed mode is less sensitive to the delay in WAN network environment. However, SNMP is more efficient for the simple network environment like LAN.

Keywords: Network management system, SNMP, and mobile agent

I. Introduction

As the public Internet and private intranet have grown from small networks into large infrastructures, the need to more systematically manage the large number of network components within these net-works has grown more important as well. In general, network management systems consist of one or more NMSs (network management stations) com-municating with network elements. The network elements can be anything that runs the common protocol suite. The software in the network element that runs the management software is called the agent. The MIB (management information base) is the database of information maintained by the agent

* 학생회원, ** 평생회원, 인하대학교 전자공학과
(Dept. of Electronic Engineering, Inha University)
※ 이 논문은 2003년도 인하대학교 지원에 의하여 연구되었음(INHA-30241-01)
접수일자: 2004년2월2일, 수정완료일: 2004년9월9일

that the manager can query or set[1].

Recently, most automated network management systems are based on the centralized approach such as SNMP (Simple Network Management Protocol) or CMIP (Common Management Information Protocol)[2]. The client/server paradigm is used in these protocols, where the NMS acts as a client with user interface and an agent behaves like a server that allows the NMS to access the agent's MIB.

In the network monitoring of SNMP, network elements accumulate network management informa-tion and send it to the manager using polling and event reporting. To convey the MIB, the polling is generally used in SNMP since the manager can query agents periodically. However, if polling is per-formed frequently, the traffic of management information can become excessively high[3]. To over-come this problem, various distributed approaches based on mobile agent have been tried nowadays[4,5].

Mobile agents can improve performance over the client/server approach. They can move to the data source, process data locally, and return with results. Performance is improved when the code to be transfe-rred is small in size compared to the number and length of messages exchanged in a traditional client /server approach. The complexity and accumulated management information can also increase the age-nt's size. Since the mobile agent visits every node, the response time can be linearly increased with accu-mulated data and the number of participants[2].

To evaluate the performance of network manageme-nts systems quantitatively, we need analytical models to evaluate the performance of the network. This paper presents analytical models of centralized approach based on SNMP protocol, distributed approa-ch based on mobile agent, and mixed mode to make up for shortcomings of SNMP and mobile agent. We compare the performance of these analytical models based on network management response time. Experi-ment results show that performance of mobile agent and the mixed mode is less sensitive to the delay in WAN network environment. This means that mobile agents are also suitable for supporting mobile and wireless users with limited bandwidths and potential costs of communication. However, SNMP is more efficient for the simple network environment like LAN.

The rest of the paper is organized as follows. In Section 2, we describe two network management paradigms, i.e., SNMP and mobile agents, and their advantages and disadvantages. In Section 3, we explain the proposed analytical models for centralized approach based on SNMP, distributed approach based on mobile agents, and mixed mode. In Section 4, we analyze the proposed models and compare the respon-se times of the three models. Finally, we give concluding remarks.

II. Related Work

2.1 SNMP

The most popular approach to management networks comes from the IETF(Internet Engineering Task Force) and is based on the SNMP (Simple Network Management Protocol). Closely related in structure is the approach based on the CMIP (Comm-on Management Information Protocol) proposed by ISO (International Organization for Standardization) for application within open interconnection networks. Both approaches assume the presence of NMSs (network management stations) that interact with management agents running on network nodes. The agents in these protocols are computational entities responsible for collecting and storing management information local to the node and responding to requests for this information from the NMS via a management protocol that specifies the packet format for a set of basic operations.

Information that is useful for network monitoring is gathered and stored by agents and made available to one or more manager systems. Two techniques are used to make the agent information available to the manager; polling and event reporting. The polling is a request-response interaction between the manager and agent. The manager can query any agent (for which it has authorization) and request the values of various information elements; an agent responds with information matching certain criteria, or supplies the manager with information about the structure of the MIB. The manager system may use polling to learn about the network configuration it is managing, to periodically obtain a change of conditions, or to inve -stigate an area in detail after being alerted by a problem. Polling is also used to generate a report on behalf of a user and to respond to specific user queries.

Client/server model introduces heavy traffic in a network. In particular, when the network is in conges -tion, the client/server model is very inefficient. While in congestion, NMS tries to reconfigure the states of network devices, which in turn makes the network more congested. Among the most note -worthy areas needing improvement is support for efficient transfer of large blocks of data and decentra -lized network management strategies. To streamline bulk data exchanges and reduce the number of trans -actions, SNMPv2 adds a new command (GetBulk), and introduces an improved version of SNMP's Get command (nonatomic Get)[6]. In the static decentrali -zed monitoring scheme in SNMPv2, there may be multiple NMSs with one system acting as a main monitoring station and the others working as area monitors to directly manage a portion of the pool of agents[7,8]. The most substantial improvement SNMP v3 offers over SNMPv1 and SNMPv2 is the addition of security features[7].

2.2 Miobile agents

The idea of employing mobile agents has been applied to applications such as distributed information retrieval, performance monitoring, and remote data filtering[9]. In the context of network management, mobile agents can improve performance over the client /server approach, if specific conditions are met. They can move to the data source, process the data locally, and return with the results. Performance is improved when the code to be transferred(i.e., the mobile agent's code) is small in size compared to the number and length of messages exchanged in a traditional client

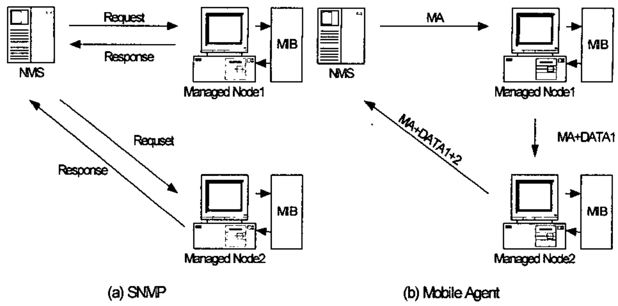


그림 1. SNMP 와 이동 에이전트
Fig. 1. SNMP and mobile agents.

/server approach. However, since the mobile agent visits every node, the response time can be increased with accumulated data and the number of partici -pants.

Figure 1 shows two network management approa -ches, i.e., centralized monitoring in SNMP and dis -tributed monitoring with mobile agents. In SNMP, there is a single NMS with which all monitored systems communicate directly. The NMS is in charge of collecting, aggregating, and processing raw net -work data. On the other hand, mobile agents visit scheduled nodes, get the local data from them, and return to NMS with the results.

2.3 Mixed mode

To reduce the number of messages that are ex -changed between the NMS and managed nodes in SNMP and to limit the amount of accumulated data that a mobile agent carries, an intermediate form of network management can be used. In this mode, the mobile agent makes tour from node to node as usual. However, rather than carrying the accumulated data along the whole path of visiting nodes, the mobile agent puts data in a temporary area of local node and visits the next scheduled node. Later, the data are sent to NMS independently by the local node. In addition, the data in temporary area can be encrypted to reinforce the security.

Data management is important to allow a mobile agent system to scale and also affects performance. this paper, this method is referred as the mixed mode and is depicted in Figure 2.

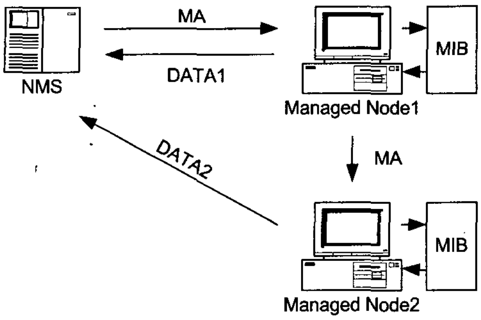


그림 2. 혼합 모드
Fig. 2. Mixed mode.

Ⅲ. Analytical Models

3.1 Network environment for analytical models

In this paper, we classify networks into two cate -gories as follows according to the location of NMS:

(1) NMS is in one network and manages only those nodes located in the same network (LAN environment).

(2) NMS is in one network and manages the all nodes located in several similar networks (WAN envi -ronment).

In the LAN environment, we call the network that is connected on Ethernet as a collision domain. We consider LAN environment to analyze the perfor -mance of network management systems in a small network. The WAN usually means a relatively large network like a company or a university network. In the case of WAN, we analyze the performance of network management systems in a large network environment.

3.2 Packet transfer in LAN

Ethernet has pretty much taken over the LAN market. Nodes in an Ethernet LAN are interconnected by a broadcast channel, so that when an adapter transmits a frame, all the adapters on the LAN receive the frame. Ethernet uses a CSMA/CD multi -ple access algorithm. On Ethernet, the client wishing to transmit must listen first to check whether there are other traffics on the medium. If there is no signal on the line, the client is free to transmit. However, if two or more clients send signals simultaneously, a

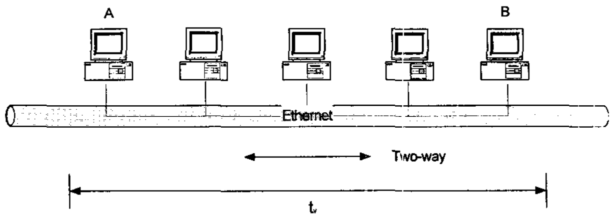


그림 3. 이더넷 네트워크
Fig. 3. Ethernet network.

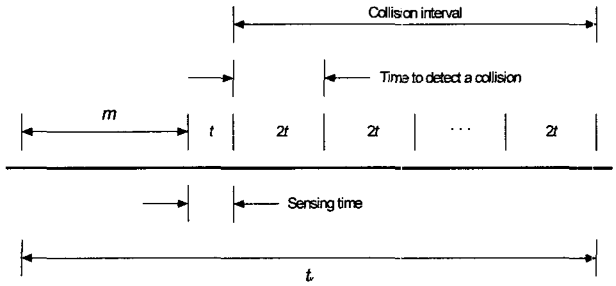


그림 4. CSMA/CD에서의 가상 전송시간 계산
Fig. 4. Calculation of the virtual transmission time in CSMA/CD.

collision occurs. CSMA/CD detects this collision state, and when it happens, the clients send signals again after waiting for a randomly selected time.

In Ethernet network, we focus attention on the two stations furthermost apart, indicated by A and B in Figure 3. We calculated the average time required to launch a message successfully. We call the time to successfully transmit a message the virtual transmi -ssion time t_v. This time has three components as shown in Figure 4. It is made of up of the time m to transmit a message, plus a time t to sense completion of transmission, plus multiples of $2t$-units of time to resolve collision, once detected. Given a collision accruing, it takes $2tN$ units of time to resol -ve the collision[10]. N represents the average number of retransmissions, given that a collision has occurred and depends on the retransmission strategy. Then virtual transmission time is given by the following equation[11].

$$\begin{aligned} t_v &= m + t + 2tN \\ &= m[1 + a(1 + 2N)] \end{aligned} \tag{1}$$

where $a = t/m$, a ratio of message transmission length m to propagation delay t.

The only way to improve the performance of

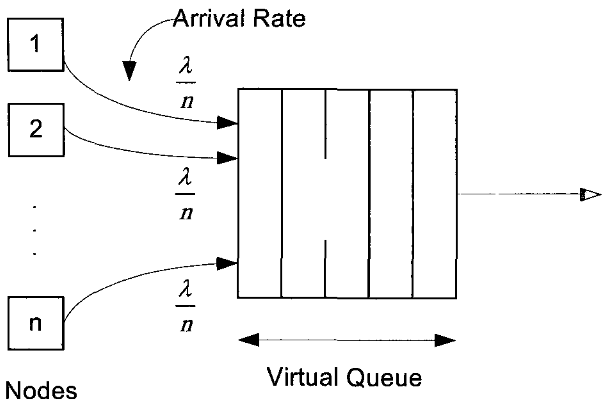


그림 5. 하나의 충돌 도메인에서의 큐잉 모델
Fig. 5. Queueing model in one collision domain.

CSMA/CD is to reduce the ratio $a = t/m$. This means either using short bus or a longer packet length. It can be easily shown that a must be limited to a small value to have the CSMA/CD strategy perform reasonably well on an average time-delay basis. As the parameter a decreases, collisions are more easily detected, and the system begins to take on the characteristics of a centralized M/G/1 queueing model as shown in Figure 5. The M/G/1 queue provides not only a limiting performance bound as $a \rightarrow 0$ for the CSMA/CD scheme, but serves to introduce a best performance bound for all multi-access strategy[10]. Then, we are able to regard one collision domain as one virtual queue. If there is no collision, the message from a source is sent without queueing, but if there is, the messages are queued and sent in a FIFO manner.

The M/G/1 queue is a single-server system with Poisson arrivals and arbitrary service-time distribu-tion. The transfer delay to transmit a frame in CSMA/CD on Ethernet can be obtained as follows[12].

$$t_f = m + \frac{\lambda \overline{m^2}}{2(1-\rho)} \qquad (2)$$

where m is an average frame transmission time (data plus overhead), λ is the total average traffic (frames/sec), and traffic intensity $\rho = \lambda m$.

If we take the average packet data length to be E(Lp) bits, and the header length is Lh bits, the parameter m is just m = [E(Lp) + Lh] / C, with C the cable transmission rate in bps. In this paper, it is assumed that the frame transmission time m has a constant value to make the analysis tractable. In this paper, [E(Lp) + Lh] is replaced with the value of MTU(maximum transmission unit) of the Ethernet.

Since the virtual queue has the Poisson distribu-tion, and the length of a frame is constant, the collision domain behaves just like M/G/1 queue. In M/G/1 queueing model, since $\overline{m^2} = m^2$ and $\rho = \lambda m$, the average delay (wait time plus transmission time) expression is given by the following equation.

$$t_f = m + \frac{\rho m}{2(1-\rho)} = \frac{m}{(1-\rho)}(1 - \frac{\rho}{2}) \qquad (3)$$

In this paper, the average transmission delay time t_f can be used as a packet transmission time between two nodes in LAN environment.

3.3 Analytical models for LAN environment

In this section, we calculate average network response time of network management systems of SNMP, mobile agent, and mixed mode using a packet transmission time t_f from equation (4). In this paper, we don't care about the process execution time, nodal overhead, and security problem.

(1) SNMP

The typical network management system offers a broad range of monitoring tools, with performance and fault reports being collected and presented. These reports are provided through the NMS issuing a regular flow of requests to agents, asking for the current value of a particular data object, conceptually part of the agent's MIB. In SNMP, the most important part is the exchange of *GetRequset* and *GetResponse* messages. Thus, we focus on those two commands in the analytical model. The NMS sends *GetRequset* message to fetch a value from a specific variable. Then, the managed node responds to the request by returning associated MIB object value with *GetResponse* message. Some common NMS tasks entail the monitoring of single MIB object, whereas others require regular updates of a number of objects.

IP Header	UDP Header	Common SNMP header	Get/Set Header	OID	Value	OID	Value	...
20 bytes	8 bytes	32 bytes						

그림 6. SNMP 메시지 포맷
Fig. 6. SNMP message format.

Figure 6 shows SNMP message format. The SNMP uses UDP (User Datagram Protocol). In gene-ral, the SNMP messages are smaller than MTU, and there is no fragmentation. The SNMP as a client /server paradigm, a network response time can be written as

$$T_{SNMP} = 2\sum_{n=1}^{N}\sum_{r=1}^{R} t_f \quad (4)$$

NMS sends requests to each node once or more in SNMP. Thus, it takes $2t_f x N x R$ time

(2) Mobile agent

The NMS unleashes mobile agent that visits each nodes and collects information locally. As mobile agent moves from a node to a node, it cumulates MIB data that it obtains and come back to NMS. When modeling such a mobile agent, we need to consider the code itself, the portion of state needed for its execution, and the portion of data that the mobile agent collects from nodes. The total size M of mobile agent can be represented as a sum of code size M_{code} of mobile agent, state size M_{state}, and cumulated data size M_{data}. For transmission of mobile agent, ATP (Agent Transfer Protocol) header is added to M, and for network transmission, TCP/IP header is added. Mobile agent size Ms can be given as follows[3].

$$M = M_{code} + M_{state} + M_{data}$$
$$M_A = ATP\_header + M \quad (5)$$
$$M_s = (TCP\_header + IP\_header) + M_A$$

When the size of mobile agent exceeds MTU (Maximum Transmission Unit), packet fragmentation occurs, and once fragmented, one more packet needs to be transferred. In this case, network response time can be calculated as follows.

$$\begin{aligned} T_{MA} = [&\sum_{i=1}^{I} N_{MA} + \sum_{i=I+1}^{2I} (N_{MA}+1) \\ &+ \sum_{i=2I+1}^{3I} (N_{MA}+2) + \cdots \\ &+ \sum_{i=(k-1)I+1}^{kI} (N_{MA}+(k-1)) \\ &+ \sum_{i=kI+1}^{N} (N_{MA}+k) \\ &+ (N_{MA}+k)] \times t_f \\ = [&\sum_{k=0}^{N/I}\sum_{i=kI+1}^{N} (N_{MA}+k) + (N_{MA}+k)] \times t_f \end{aligned} \quad (6)$$

where $N_{MA} = M_S / MTU$ and means an initial number of fragmented mobile agents. As mobile agent gets data while visiting, the message size will grow and, if the size reaches MTU value, the mobile agent will be fragmented. I represents the maximum number of nodes that the mobile agent can visit without fragmentation and can be obtained from $I = MTU / MIB_{value}$, where MIB_{value} is the size of MIB object that mobile agent puts when it visits nodes.

(3) Mixed mode

In the mixed mode, while mobile agent travels through managed nodes, local data it gets are saved in temporary area and sent to NMS via TCP/IP later. Network response time T_{Mixed} of mixed mode can be obtained from the following equation.

$$T_{Mixed} = \sum_{n=1}^{N} N_{MA} \times t_f + \sum_{n=1}^{N} t_f \quad (7)$$

In this mode, there is no need to add local data obtained from each node that it visits like mobile agent, but it may need additional time for the mana-ged node to send these data to NMS later. In the above equation, the second term accounts for this time.

3.4 Analytical models for WAN environment

In WAN environment, managed nodes can be scat-tered in different collision domains. Thus, we should consider bandwidth and delay time between these domains. Figure 7 shows an example of WAN topology. In the figure, transit AS (autonomous sys-tem) is assumed to be located in the middle of

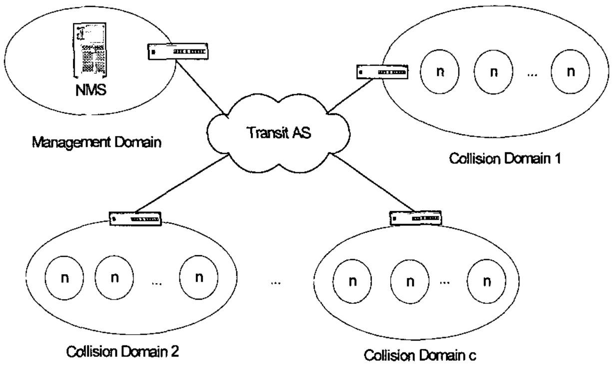


그림 7. WAN 네트워크 환경
Fig. 7. WAN network topology.

collision domains so that we can obtain the expre -ssion to calculate network response time considering the delay time between domains[10]. We call the do -main that the NMS resides the management domain.

(1) SNMP

In SNMP, all request and response messages go through the transit AS. Specifically, a request mess -age starts from the management domain, goes through the transit AS, and gets to a managed node in a collision domain. Response messages go through in reverse order.

Network response time T^W_{SNMP} of SNMP can be calculated as follows.

$$T^W_{SNMP} = T_{SNMP\cdot M} + T_{SNMP\cdot 1} + T_{SNMP\cdot 2} + \cdots + T_{SNMP\cdot c} + (N \times 2D) \tag{8}$$

where N is the number of nodes in a collision domain, and D is the delay time in the transit AS. The total network response time is made of packet transfer delay T_{SNMP_M} in management domain, pack -et transfer delay $T_{SNMP\_1} + T_{SNMP\_2} + \cdots + T_{SNMP\_M}$ in collision domains, and packet delay $N \times 2D$ in the transit AS. T_{SNMP_M}, T_{SNMP_1}, T_{SNMP_2}, ... , and T_{SNMP_c} can be calculated from equation (4). Equation (8) shows that SNMP is sensitive to the delay of transit AS since all request and response messages go through the transit AS.

(2) Mobile agent

To get an analytical model in WAN, one management domain and one collision domain are consi dered first. Network response time $T^W_{MA\cdot 1}$ of mobile agent can be written as follows.

$$T^W_{MA\cdot 1} = N_{MA}(t_f + D) + T_{MA\cdot 1} + (N_{MA} + k)(t_f + D) \tag{9}$$

In the above equation, $N_{MA}(t_f + D)$ is the time for a mobile agent to move from the NMS to a collision domain, and T_{MA_1} is the response time in the colli -sion domain and obtained from equation (5). When mobile agent comes back to NMS after finishing requested tasks, it takes $(N_{MA} + k_1)(t_f + D)$ unit time, where k_1 is an additional number of packets genera -ted from the fragmentation of mobile agent.

Extending to two collision domains, network res -ponse time $T^W_{MA\cdot 2}$ of mobile agent can be obtained from the following equation.

$$\begin{aligned} T^W_{MA\cdot 2} = {} & N_{MA}(t_f + D) + T_{MA\cdot 1} \\ & + (N_{MA} + k_1)(t_f + D) \\ & + T_{MA\cdot 2} + (N_{MA} + k_2)(t_f + D) \end{aligned} \tag{10}$$

In the above equation, $(N_{MA} + k_1)(tf + D)$ account for network delay time when a message moves from the first domain to the second one through transit AS. Likewise, after mobile agent completes its job in the second domain, it takes $(N_{MA} + k_2)(t_f + D)$ unit time to move from the second domain to the manage -ment domain. If the total number. of nodes in a domain is the same in both the first and second domains, k_2 is equal to k_1, which is also used in equation (9). If we generalize the above procedure to c collision domains, network response time T^W_{MA} of mobile agent can be written as follows.

$$T^W_{MA} = N_{MA}(t_f + D) + \sum_{i=1}^{c} T_{MA\cdot i} + \sum_{i=1}^{c} (N_{MA} + k_i)(t_f + D) \tag{11}$$

In case of SNMP, since NMS exchanges request and response messages with every node, and these messages must go through the transit AS, the net -work response time depends on the number of managed nodes. On the other hand, the network response time of mobile agent approach depends on the number of collision since the transit AS is traversed only when the mobile agent moves from one domain to another. The size of mobile agent grows when it collects information from each visited node, and the message is fragmented if the size exceeds the MTU. In this case, all fragmented messages also have to traverse the transit AS. Thus, the network response time of mobile agent also depends on the mobile agent size

(3) Mixed mode

To derive an analytical model for mixed mode, we first consider one management domain and two collision domains. Network response time $T^{W}_{Mixed\,2}$ of mixed mode is given as follows.

$$T^{W}_{Mixed\,2} = N_{MA}(t_f + D) + T_{Mixed\,1} + N_{MA}(t_f + D) + T_{Mixed\,2} + \sum_{n=1}^{N} (t_f + D) \quad (12)$$

where the first $N_{MA}(t_f + D)$ represents message delay time when the mobile agent of mixed mode moves from NMS to the first collision domain, and the second one accounts for message delay time when the mobile agent moves from the first collision domain to the second one. In each domain, we can get message delay time using equation (7) for T_{Mixed}. The last term in the above equation represents the time for each managed node to send local data to NMS.

Generalizing to c collision domains, network response time T^{W}_{Mixed} of mixed mode can be obtained as follows.

$$T^{W}_{Mixed} = \sum_{i=1}^{c} N_{MA}(t_f + D) + \sum_{i=1}^{c} T_{Mixed\,i} + \sum_{n=1}^{N} (t_f + D) \quad (13)$$

As shown in the above expression, network response time of mixed mode is dependent on the number of nodes and the number of collision domains. This shows that network response time of mixed mode is less sensitive to the transit AS delay compared to SNMP, but more sensitive to the transit AS delay compared to the mobile agent.

Since the information obtained from each node is directly sent to NMS by the local node, the size of mobile agent does not grow. Thus, the network response time is insensitive to the kind of tasks performed by the mobile agent and the kind of the objects that the mobile agent obtains.

IV. Analyses of Analytical Models

4.1 Analyses of analytical models in LAN

We compare the performance of three analytical models, i.e., centralized approach based on SNMP protocol, distributed approach based on mobile agent, and mixed mode. In this section, we evaluate the formulas for network response time of these approaches under LAN environment. In this environment, all of the nodes including NMS are located in the same collision domain. Table 1 shows common parameter values used in this evaluation.

The number of managed nodes is N = 248, and the transmission rate of Ethernet network is C = 10Mbps. The message transmission length m can be calculated from $m = (MTU \times 8)/C$, and ρ is set to constant value of 0.5. Note that these values are used in calculating transmission delay time t_f .

In our simulations, we suppose the request and response messages of SNMP are not fragmented since they usually cannot exceed MTU. Besides, we assume there is no link error on the network, and we don't consider application processing time in each node.

표 1. 공통적으로 사용된 파라미터 값
Table 1. Common parameters.

Parameter	Value
N	248
C	10Mbps
MTU	1500Byte
ρ	0.5
m	$(MTU \times 8)/C$

To evaluate network response time for SNMP, mobile agent, and mixed mode in LAN environment, we consider various values of parameters such as the size of MIB object (MIB_{value}), mobile agent size, and the number of repeated requests to each node (R). First, we evaluate the network response time when MIB_{value} is equal to 40Byte, R is 1, and mobile agent size is 4.5Kbyte. Figure 8 shows the network respon-se time for three methods. The result reveals that SNMP has the best response time compared to the other two methods under LAN environment. This is because all nodes are connected within a single collision domain and SNMP can perform management operations with one small packet. On the contrary, mobile agent and mixed mode have difficulties in traveling all of the nodes since initial packet size of mobile agent and mixed mode is larger than SNMP, which slows down the movement of the packets. From the figure, we know that the response time of SNMP and mixed mode increases linearly with the number of nodes, while mobile agent does exponentially.

To show the effect of the number of requests on network response time, we use a larger number for R in Figure 9 and assume that R = 3. Figure 9 shows the graph of simulation results in LAN envi-ronment. In case of management task like Interface utilization[10], the NMS in SNMP repeatedly communi-cates 3 times with each node. In the mobile agent or mixed mode, these three operations are performed all together when the mobile agent visits each node. Thus mobile agent or mixed mode does not require additional communications compared to SNMP. How-ever, the mobile agent accumulates MIB objects for those repeated operations, which results in packet fragmentation. Therefore, we observe the worst performance for the mobile agent. It is clear that the mixed mode has the shortest network response time since each node directly transfer its MIB objects to the NMS. Thus the mixed mode is minimally affected by repeated requests. Note that, when the number of network entities is fewer than 100, the mobile agent has shorter response time than SNMP. However, as the number of nodes grows over 100, the network response time of mobile agent becomes longer than that of SNMP.

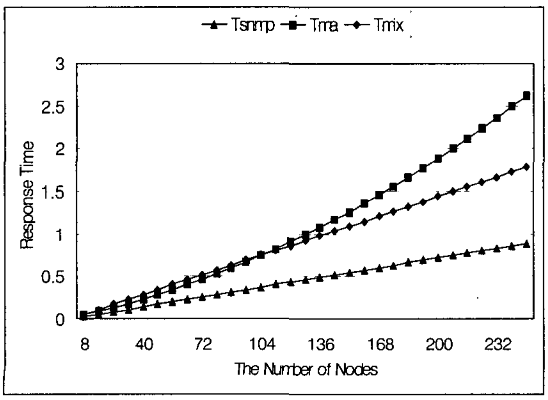


그림 8. 관리 노드 수에 따른 전체 응답시간
Fig. 8. Network response time in LAN.

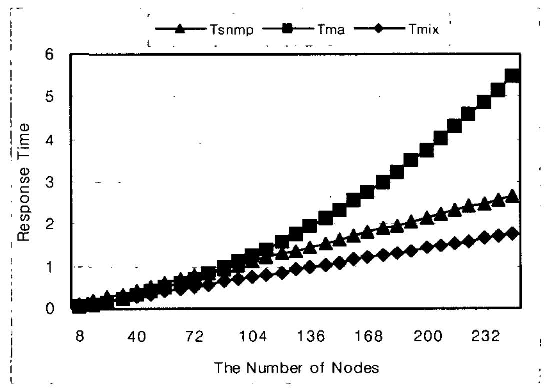


그림 9. 각 노드로 데이터 요구 회수가 3회인 경우의 전체 응답시간
Fig. 9. Network response time when the number of request to each node is 3.

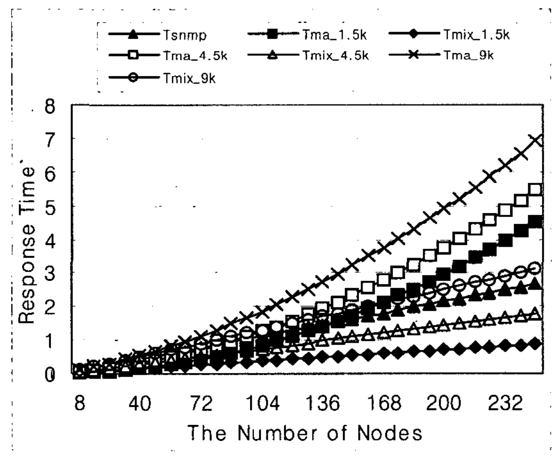


그림 10. 이동에이전트 크기에 따른 응답시간
Fig. 10. Network response time with the various size of data in mobile agent.

Next, we use various sizes of the mobile agent from 1.5 Kbyte to 9 Kbyte. Figure 10 shows the result when the number of repeated requests to each node is 3 in LAN environment. When the mobile agent size is less than 4.5Kbyte and the number of nodes is fewer than 100, the network response time of the mobile agent and mixed mode shows better performance compared to T_{SNMP}. Therefore, if we design network management system with the mobile agent or the mixed mode, it's recommended to limit the number of managed nodes under 100 in one collision domain.

Last, we use various values of MIB object size MIB_{value} to show how the MIB object size affects the network response time of SNMP, mobile agent, and mixed mode in LAN environment. The simulation results are shown in Figure 11 when we fix the mobile agent size. As the size of MIB_{value} increases, the network response time of mobile agent drama -tically increases. In short, the network response time of the mobile agent is very sensitive to the MIB object size.

4.2 Analyses of analytical model in WAN

In this paper, we separate the WAN environment into two categories, i.e., internet and intranet, accor -ding to the transit AS delay time. Intranet usually has a small transit AS delay while internet has a relatively long transit delay. In our simulations, the transit AS delay for intranet has the value less than 2ms. But the transit AS delay for internet is longer that 2ms. The other parameters have the same as in the LAN environment.

In the LAN environment, NMS and managed nodes are located in the same domain. However, in WAN environment, NMS and managed nodes are scattered among different domains.

In Figure 12, we compare LAN environment with intranet environment. We set transit AS delay to 1ms for intranet environment. As this and the previous graphs show, SNMP has the best performance in LAN environment. On the other hand, even with a small increase of the delay time, the network response time of SNMP in intranet environment is dramatically increased. However, the response time of mobile agent shows just a little increment compared with SNMP.

To figure out the effect of transit AS delay to network response time of SNMP, mobile agent, and mixed mode in WAN environment, we vary the transit AS delay from 1ms to 10ms. As Figure 13 shows, SNMP is very sensitive to the transit AS delay because all of the request and response messages go through the transit AS. Since the mobile agent is not affected by transit AS delay, it is reasonable to design network management system with mobile agent when managed nodes are scattered among several collision domains and is far apart from NMS with a relatively large delay. In case of mixed mode, the network response time of mixed mode also

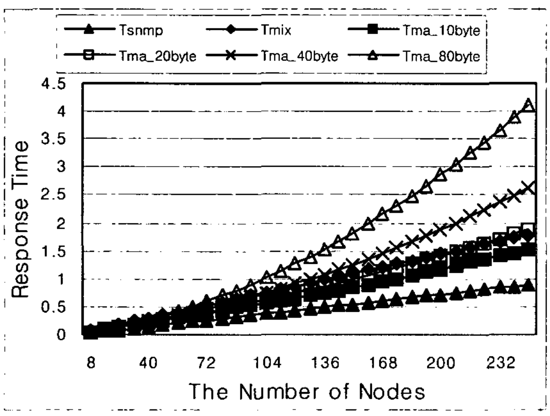


그림 11. 태스크에 따른 응답 시간 비교
Fig. 11. Network response time for various values of MIB object size.

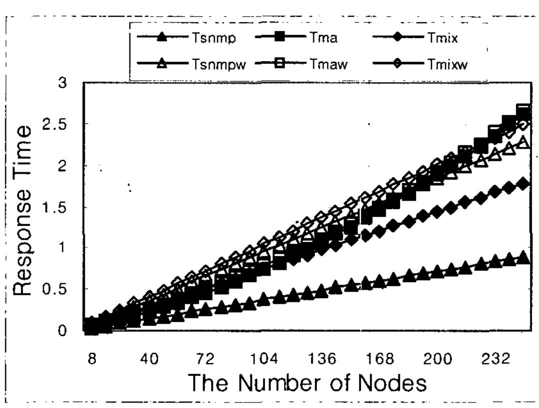


그림 12. LAN vs WAN 환경의 응답시간 비교
Fig. 12. Comparison of response times; LAN versus intranet.

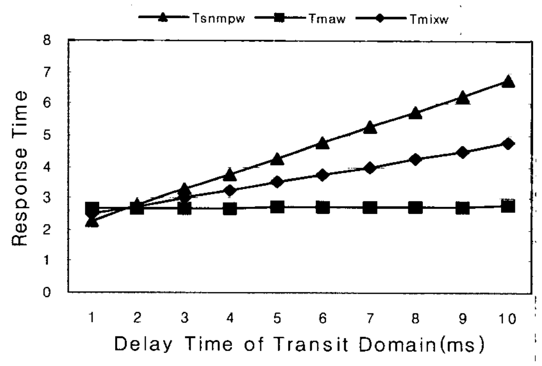


그림 13. 경로 AS의 지연 시간에 따른 응답시간
Fig. 13. Network response time for various transit AS delays.

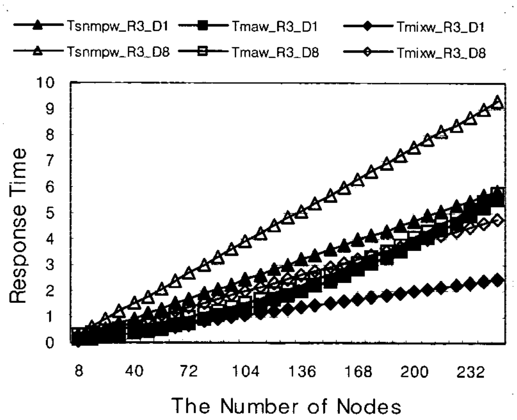


그림 14. 태스크와 지연에 따른 응답 시간 비교
Fig. 14. Network response time for various.

increases since all response messages go through the transit AS when they are delivered from each node to NMS. However, unlike SNMP, the network response time of mixed mode is in between those of SNMP and mobile agent because only response messages go through the transit AS.

Figure 14 shows the simulation results when we vary the transit AS delay, and each node is repeatedly visited three times. When the transit AS delay is 1ms, the mixed mode has better performance than the mobile agent. This result is caused by relatively constant delay time of the mixed mode. However, the mobile agent generates message frag-mentation that increases the network response time. When the transit AS delay is 8ms and the number of nodes is fewer than 150, the network response time of mobile agent is better than mixed mode. As a result, the mobile agent is seriously affected by the MIB object size. And the mixed mode is more sensi-tive to the transit AS delay time than the mobile agent. Meanwhile, since SNMP shows the worst performance for all simulations under the WAN envi-ronments, it is not suitable idea for network management method in those kinds of network environments.

V. Conclusion

This paper presents analytical models of centralized approach based on SNMP protocol, distributed approach based on mobile agent, and mixed mode to make up for shortcomings of SNMP and mobile agent. We compare the performance of these analy-tical models based on network management response time.

Experiment results show that performance of mobile agent and the mixed mode is less sensitive to the delay in WAN network environment. However, SNMP is more efficient for the simple network environment like LAN.

References

[1] Stallings, SNMP, SNMPv2, SNMPv3, and RMON1 and 2, 2nd Edition, Addison-Wesley, 1999.

[2] V. A. Pham and A. Karmouch, "Mobile software agents: An overview", IEEE Communications Ma-gazine, vol. 36, July 1998.

[3] M. Baldi and G. P. Picco, "Evaluation the tradeoffs of mobile code design paradigms in net-work management applications," ICSE'98, Apr. 1998.

[4] A. Sahai and C. Morin, "Towards distributed and dynamic network management," Proc. of the IEEE/IFIP Network Operation and Management Symposium, Feb. 1998.

[5] Susilo, A. Biezczad, and B. Pagurek, "Infrastruc-ture for advanced network management based on mobile code," Proc. of NOMS'98, Feb. 1998.

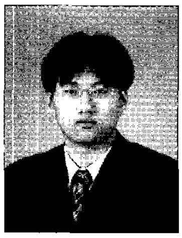
[6] W. Stallings, "SNMP and SNMPv2: The infrast-ructure for network management", IEEE Com-munications Magazine, vol. 36, March 1998.

[7] A. Liotta, G. Pavlou, and G. Knight, "Exploiting agent mobility for large-scale network monito

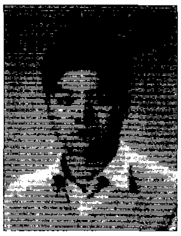
-ring", Network, IEEE, vol. 16, May 2002.
[8] D. Case, K. McCloghrie, M. Rose,and S. Wald-busser, "Structure of management information for version 2 of the Simple Network Manage-ment Protocol," RFC 1902, Jan. 1996.
[9] Fuggetta, G. P. Picco, and G. Vigna, "Understan-ding code mobility," IEEE Transactions on Soft-ware Engineering, vol. 24, May. 1998.
[10] W. Zegura, K. L. Calvert,and M. J. Donahoo, "A quantitative comparison of graph-based models for internet topology," IEEE/ACM Transaction on Networking, Dec. 1997.
[11] Schwartz, Telecommunication networks: Proto-cols, modeling and analysis, Addison-Wesley, 1987.
[12] Kleinrock, Queueing systems volume 1: Theory, John Wiley & Sons, 1975.

저 자 소 개

이 정 우(학생회원)
1992년 인하대학교 전자공학과 학사.
2003년 인하대학교 정보통신대학원 석사.
<주관심분야: 컴퓨터 네트워크, 네트워크매니지먼트 시스템, 신호처리>



최 재 혁(학생회원)
2001년 인하대학교 전자공학과 학사.
2004년 인하대학교 전자공학과 석사.
2004년~현재 인하대학교 전자공학과 박사과정.
<주관심분야: 컴퓨터구조>



이 우 석(학생회원)
2003년 인하대학교 전자공학 및 컴퓨터공학과 학사.
2003년~현재 인하대학교 전자공학과 석사과정.
<주관심분야: 임베디드 시스템, 마이크로프로세서, SoC 아키텍쳐>

최 상 방(평생회원)
1981년 한양 대학교 전자공학과 학사.
1988년 University of Washington 석사.
1990년 University of Washington 박사.
1981년~1986년 LG 정보통신(주).
1991년~현재 인하대학교 전자공학과 교수.
<주관심분야: 컴퓨터구조, 컴퓨터 네트워크, 병렬 및 분산처리 시스템, Fault-tolerant computing>