論文92-29A-7-2

표준망관리 프로토콜의 Scope/Filter에 대한 성능분석

(A Performance Analysis on the Scope/Filter of OSI Network Management Protocol)

邊 玉 煥*, 陳 庸 玉**

(Ok Hwan Byeon and Yong Ohk Chin)

要 約

본 논문에서는 표준망 관리 프로토콜인 CMIS/CMIP에서 시스템과 망의 성능에 영향을 주는 범주결정/필터 파라메터를 대상으로 시뮬레이션 모델을 정하고, 성능분석을 하여 망관리 시스템에서의 최적 폴링타임, 그리고 대기행렬 용량과 요구응답시간의 비교분석을 통해 적정한 범주형태를 제시하였다. M-Get, M-Set, M-Action, M-Delete 서비스 등에 사용되는 범주결정/필터 파라메터의 임의의 피관리객체(Managed Object)를 선택하는 방법을 제공하며, 요구한 범주형태 및 필터링 결과에 따라 시스템 및 망에 미치는 영향은 크게 변동하게 된다. 망의 부하와 폴링타임을 변형하면서, 각 범주형태 별, 그리고 망에서 모든 범주형태를 동시에 이용하는 경우로 나누어 시뮬레이션하였다. 이때 요구응답시간, 관리시스템의 대기행렬용량, 지연시간을 측정하였다. 그 결과 표준망 관리에서 망의 성능을 향상시키는 범주형태와 폴링타임을 제시할 수 있었다.

Abstract

In this paper, a simulation model for scope and filter parameters in CMIS/CMIP, OSI management protocol, has been discussed and the suitable scope type was represented by analyzing optimal polling time in network management system, queue size and response time. Scope and filter parameters in M-Get, M-Set, M-Action, and,= M-Delete services provide a scheme to select some managed objects, and the results are largely varied according to a scope type requested and the outcome of filtering. It has been simulated each scope type and all scope type mixed in a network by varying the offered load and a polling time. On the same time, the response time, queue size of the managing system and a delay time have been measured. As a results, the suitable scope type and polling time enhancing the network performance on the OSI network management is proposed.

I. 서 론

최근 컴퓨터와 통신기술의 발전으로 대규모 전산망이 형성되고, 다양한 기능을 제공해주는 분산처리 운영환경의 확대 등으로 인해 전산망을 사용하고자 하는 요구와 전산망내의 정보통신량이 급격히 증가되고 있다.

이러한 사용자의 요구와 통신기술의 발전에 부응하여 양질의 정보통신서비스를 제공하기 위한 통신망, 통신프로토콜 및 응용프로그램 등은 점점 복잡해지고 다양해져가는 추세이다. 따라서 이러한 전산망에 최신의 정보통신 서비스를 어떻게 제공할 것인가

*正會員, 시스템 工學 研究所

(System Eng., Research Institute)

**正會員, 慶熙大學校 電子工學科

(Dept. of Elec. Eng., Kyunghee Univ.)

接受日字 : 1992年 3月 7日

하는 문제와 더불어, 전산망을 안전하면서도 효율적으로 운영하기 위해 정보통신 서비스를 제공하는 통신자원을 어떻게 관리하느냐 하는 문제가 급격히 부상하게 되었다.[1,2,3]

일반적으로 관리의 개념은 망 운영에 필요한 모든 정보를 망을 구성하는 다양한 통신장비 및 전송설비로부터 수집하여 망이 효율적으로 동작하도록 감시, 제어 및 보고하는 기능의 수행을 총칭한다. 전산망에서는 이러한 기능을 수행하는 망 관리 시스템들간에 표준화된 정보를 주고 받음으로 해서 유기적으로 전체 망을 관리하는 형태로 발전해가고 있다.[4,5,6] 현재 ISO에서는 통신장비나 관리형태에 무관하게 전체 망을 통합된 망 관리시스템을 이용하여 관리할 수 있도록 OSI관리 표준을 제안하고 있다.[1,2]

OSI관리는 OSI환경내의 여러자원의 제어, 조정 및 감시를 위한 메카니즘과 이들 자원에 관한 정보를 전송하기 위한 OSI관리 프로토콜 표준을 규정하고 있다.[7,8,13] 또한 OSI 관리에서는 OSI 환경내에 있는 관리하고자 하는 자원들에 대한 관리동작(management operation)과 자원으로-터의 사건통지(event noitification)를 표현하기 위해 이들 자원을 추상적으로 표현하여 어떤 특성을 가진 관리객체(managed object)로 표현하며, 개방 시스템내에 있는 관리정보는 MIB(management information base)라는 개념적인 장소에 저장된다.[10,11,12,13] 이러한 OSI 관리의 표준화는 자원을 관리하기 위해 적용되는 다양한 기능을 표준에 맞게 공통적으로 정의하는 부분과, 시스템간의 정보교환을 위한 통신수단을 제공하는 부분 및 자원에 적용되는 특정한 관리동작과 정보등을 정의하는 부분등 크게 세가지로 나누어 볼 수 있다.

본 논문에서는 이들 중 시스템간의 관리정보 교환을 위한 서비스와 프로토콜을 제공해주는 공통관리 서비스 및 프로토콜(CMIS/CMIP:common management information service/ protocol)[7,8]을 중점적으로 연구하여 관련 파라메터중 전산망의 성능에 많은 영향을 주는 범주결정/필터 파라메터의 변화에 따른 적정 범주형태등 전산망 성능에 미치는 영향을 분석하고 그 특성을 제시하고자 한다. 망관리에 대한 연구는 주로 관리구조, 관리기능들을 중심으로 진행되고 있고 망관리 프로토콜과 scope/filter등 관련 파라메터에 대한 전산망 성능분석은 크게 진척되고 있지 않다.[17,18] 성능분석 분야는 관리기능을 감안하지 않은 일반 통신 프로토콜에 대한 분석이 주로 이루어져 왔다.[14]

본 연구에서는 Mil3 Inc.의 시뮬레이션 도구인 OPENT[16]을 이용하여 Ethernet상에서 시스템과 망 성능에 영향을 주는 CMIS/CMIP의 범주결정/필터 파라메터를 대상으로 시뮬레이션 모델을 정하고 성능분석을 하여 망관리시스템에서의 최적 폴링타임, 그리고 대기행렬용량과 요구응답시간의 비교 분석을 통해 적정 범주형태를 제시하였다. 2장에서 표준 망관리 프로토콜의 구조 및 동작고찰, 3장에서 시스템의 시뮬레이션 모델설정, 4장에서는 실험의 결과를 비교분석하고 마지막으로 본 논문의 결론 및 앞으로의 연구방향에 대해 논한다.

Ⅱ. 표준망관리 프로토콜 구조 및 동작

OSI관리모델은 그림1에서와 같이 객체가 분산되어 있는 환경에서 일종의 객체지향 모델(Object oriented Model)이며, 관리프로토콜이란 그 분산된 객체간의 정보교환으로 볼 수 있다.[1,2,5] OSI관리기능을 수행하기 위해서 관리동작과 사건통지를 교환하는 관리 응용 프로세스를 관리 정보서비스(MIS:management information service) 이용자라 부르고 관리자(manager)와 피관리자(agent)로 분류하여 정보교환을 수행한다.

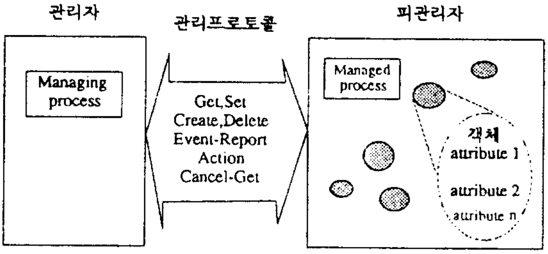

그림 1. OSI관리모델
Fig. 1. OSI management model.

관리자는 관리동작을 발하고 이에 대한 사건통지를 받는 응용프로세스이며 피관리자란 관리동작을 수신하고 이를 실행한 다음 그것에 대한 사건통지를 발행하는 응용프로세스이다. 전산망에서 관리대상이 되는 관리객체는 추상적으로 표현되며 해당영역을 담당하는 피관리자에 의해 감시된다. 원격으로 관리자에게 사건통지가 전달되면 관리자가 사건을 인식하게 되므로 관리객체가 관리자로부터 항상 원격으로 폴링에 의해 감시되는 모델은 아니다. 따라서 관리동작에 대한 실행책임은 관리자에게 있는 것이 아니고 피관리자에게 있게 된다.[5,9]

1. 공통관리정보서비스(CMIS:common management information service)

CMIS/CMIP은 OSI 관리체계[6]에서 정의된 바와 같이 시스템 관리를 위하여 응용프로세스에 의해 사용되는 응용계층 서비스 및 프로토콜로 관리정보모델에서 정의된 관리정보의 동작과 통지에 대응하는 관리동작 그리고 그 결과를 개방시스템간에 요청하거나 보내는데 사용된다.

CMIS는 시스템 관리를 위한 정보나 명령어들을 교환하기 위해 개방형 시스템의 응용프로세스에 의해 사용되며 관리동작 서비스 및 관리통지 서비스가 있다.[7] 그리고 부가적인 기능으로 다중응답(linked multiple response), 다중객체 선택(multiple object selection), 필터링(범주 결정 기능과 연계) 등이 제공된다.[7,8] 이들중 관리동작 서비스는 관리자가 피관리자로 하여금 관리객체에 대한 감독과 제어에 관한 동작을 하도록 요구하는 서비스를 제공하며 표1과 같은 동작이 지원된다. 그리고 표2에는 이들중 M-GET 서비스의 파라메터를 보여주고 있다.

표 1. OSI관리 동작 서비스
Table 1. OSI management operation service.

관리동작서비스	내 용
M-GET	관리객체에 대한 정보검색을 지원
M-SET	관리객체에 대한 파라메터값을 변경
M-ACTION	관리객체의 특정한 행동을 수행하도록 요구
M-CREATE	관리객체의 생성요구
M-DELETE	관리객체의 삭제요구
M-CANCEL-GET	이전에 발해졌던 M-GET서비스의 취소요구

표 2. M-GET 관리동작서비스 파라메터
Table 2. The parameters of M-GET management operation service.

파라메터 명	Req/Ind	Rsp/Cnf.
Invoke identifier	M	M
Linked identifier	–	C
Base Object Class	M	–
Base Object instance	M	–
Scope	U	–
Filter	U	–
Access control	U	–
Synchronization	U	–
Attribute identifier list	U	–
Managed object class	–	C
Managed object instance	–	C
Current Time	–	U
Attribute list	–	C
Errors	–	C

*M:Mandatory U:service-user option C:Conditional

2. 공통관리 정보프로토콜(CMIP:common management information protocol)

CMIP는 관리정보를 교환하기 위해 시스템 관리 응용 엔티티가 사용하는 프로토콜로서 CMIS에서 정의하는 동작서비스와 통지 서비스를 제공하기 위해서는 몇개의 서비스 요소들이 필요한데 이러한 요소들과 상대 엔티티 사이에 교환되는 PDU들을 규정하는 프로토콜이다. 그림2는 CMIP를 실현하는 응용 엔티티인 시스템관리응용엔티티(SMAE:system management application entity)의 논리구조를 나타내고 있다.[4,8,13,15]

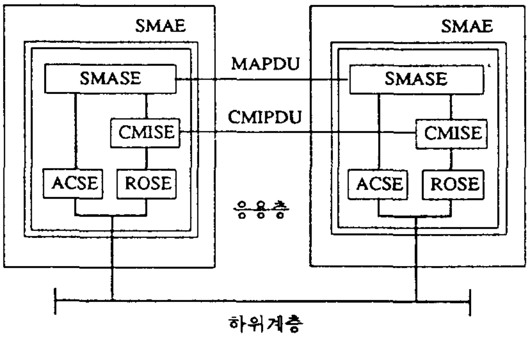

그림 2. SMAE 논리구조
Fig. 2. The logical structure of SMAE.

시스템 관리응용서비스요소(SMASE)는 상위관리 프로세스에게 시스템관리기능을 제공하고 CMISE는 관리동작과 통지를 교환하기 위한 수단을 제공하는 응용서비스 요소이다. CMIP를 지원하기 위한 하위계층서비스로는 원격동작을 위해 정의된 RO-INVOKE, RO-RESULT, RO-ERROR, RO-REJECT-U, 그리고 RO-REJECT-P 서비스를 사용하며 연결제어를 위해 A-ASSOCIATE, A-RELEASE, A-ABORT 및 A-P-ABORT 서비스를 사용한다. CMIP 관리절차는 ASN.1으로 표기된 CMIS서비스 프리미티브들의 구조, 코딩 및 관련성에 대한 CMIP PDU의 전송에 대해 규정하고 있으며 여기에는 연결확립, 연결종료, 원격동작, 사건보고, GET, SET, ACTION, CREATE, DELETE, CANCEL-GET의 10가지 서비스들에 대한 절차가 있다.

CMISE 이용자와 CMISE 제공자간의 프로토콜인 CMIP 동작절차를 개략적으로 표현하면 다음과 같다.[7,8]

(1) CMISE 이용자는 제공자에게 관리기능에 필요한 서비스를 호출하고 그에 따른 파라메터를 제공한다.
(2) CMIPM(common management information protocol machine)은 각 서비스 요청에 대해 필요한 PD PDU를 구성한다.
(3) CMIPM은 관리정보전송을 위해 ROSE의 서비스를 이용하여 원격의 CMIPM에게 PDU를 전달한다.
(4) 수신 CMIPM은 ROSE로 부터 전달된 PDU의 PCI(protocol control information)를 분석한다.
(5) CMIPM은 CMIP 이용자에게 적절한 서비스 프리미티브를 통해 전달한다.
여기서의 CMIPM은 CMIS요청 및 응답 프리미티브를 받아 CMIP PDU를 생성함으로서 CMIP의 특정 절차를 시작하게 하는 기능을 가진 프로세스이다[8]

3. 피관리 객체(Managed Object)의 선택 범위

관리자와 피관리자 사이에 관리정보 교환을 위해서는 우선 어떤 범위의 정보를 가져올 것인가를 알아야 하는데 이는 표2에서 보여주는 범주결정과 필터 파라메터를 적절히 선정하여 수행할 수 있다. 이들은 각각 관리행위에 필요한 피관리 객체의 범주를 결정하는 것과 이 선정된 피관리객체의 범주에서 꼭 필요한 피관리객체들을 찾아내기 위한 것이다. 범주결정과 필터 파라메터의 사용을 가능하게 하려면 복수 객체 선택(Multiple Object selection) 기능요소와 복수응답(multiple reply) 기능요소 그리고 필터기능 요소의 사용이 필요하다[7,8,17] 범주결정과 필터 파라메터는 관리 데이타의 양을 조절할 수 있는 파라메터이므로 이의 적절한 선정이 망의 전체적인 성능과도 밀접한 영향이 있을 것임을 알 수 있다.

(1) 범주결정(Scoping)

범주결정은 관리정보트리(MIT:management information tree)에 있는 피관리 객체들을 선정하기 위한 작업이다. OSI에서 규정하고 있는 관리정보트리의 기준점은 관리자에서 주어지는 기준 피관리객체(base managed object)에 의해 결정된다. 즉 기준 피관리객체는 MIT에서 선정된 서브트리의 루트가 되며 여기서부터 범주결정 및 필터링을 위한 검색이 시작된다. 이를 중심으로 전개되는 범주결정 레벨은 필터기능이 적용될 레벨을 말하는데 다음과 같이 4가지 경우들이 있다.

- 기준 피관리 객체 자신(type1)
- 기준 피관리객체 아래의 n번째 레벨(type2)
- 기준 피관리 객체로 부터 이 아래의 n번째 레벨을 포함한 모든 객체(type3)
- 기준 피관리 객체로부터 이 아래의 모든객체(type4)

(2) 필터링

관리정보의 범주가 결정되면 선정된 피관리객체들에 대해 특정조건을 만족하는 경우만을 선택하는 필터링 기능이 적용될 수 있다. 이 기능은 결정된 피관리객체들에 대해서 속성값이나 이의 존재유무등에 대한 조건들을 규정한 것이라고 할 수 있다.

Ⅲ. 시스템의 시뮬레이션 모델

범주결정과 필터 파라메터에 대한 시스템의 모델링은 기능적으로 2가지로 양분된다. 즉, 관리시스템과 피관리시스템이 임의의 망에서 상호 동작하게 된다. 기본 망은 Ethernet을 가정하였으며 하나의 관리시스템이 다수의 피관리시스템을 관리하도로 모델링 하였다. 이때, 사용되는 시뮬레이션 도구는 OPNET이라는 시뮬레이션 전용 소프트웨어를 이용하였다. OPENT은 3단계로 시뮬레이션 모델을 설정하는데 이는 네트워크 모델, 노드 모델, 프로세스 모델로 구성된다. 네트워크 모델은 Ethernet상에 30개의 노드로 구성되도록 하였으며 노드모델은 네트워크 모델의 구성원 즉, 시스템에 대한 모델이다. 마지막으로 프로세스 모델은 각 노드 내부에서 운용되는 프로토콜, 메카니즘, 알고리듬 등을 유한상태기계로 모델링 된다. 시스템에 대한 시뮬레이션 모델은 그림3과 같다.

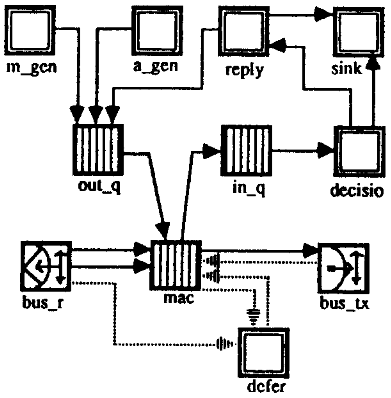

그림 3. 관리시스템 모델
Fig. 3. Management system model.

관리시스템은 Ethernet상의 피관리시스템을 관리하기 위한 관리트래픽과 관리와 관계없는 일반트래픽으로 분류되어 메시지를 발생시킨다. 발생 패킷의

크기는 상수로 1Kbits를 발생시키며 목적지 주소는 자신을 포함하여 난수를 발생하여 사용한다. 관리모델의 경우 M-GET등과 같은 패킷은 기본적으로 피관리시스템의 관리객체에 대한 검색이므로 주기적으로 피관리시스템에게 문의하도록 구성되었다. 관리요청 패킷내에는 범주에 대한 정보를 첨가하여 피관리시스템은 이를 기초로 자신의 관리정보트리에서 관리객체의 범주를 결정하고 필터링을 행하도록 한다. 범주에 대한 정보는 범주 형태와 관리정보 트리의 레벨로 나뉘어 지는데 이 값은 난수를 발생시켜 적용한다. a-gen 모듈에서는 일반 패킷을 발생시키는데 패킷간의 발생시간 $t_i(x)$는 지수분포를 따르며 평균 발생횟수를 나타내는 $1/\lambda$은 0.5로 한다. 송수신의 완충장치로 대기행렬을 이용하며 서버를 대기행렬 시스템 내부에 갖고있다. 대기행렬의 크기는 각각 시뮬레이션시 무한대로 하였으며 평균서비스율 μ는 128PPS(packet per second)이다. 유입되는 패킷은 in-q 모듈을 경유하여 decision모듈에 이르게 되는데 이 모듈에서는 패킷의 형태에 따라 일반패킷인 경우 sink 모듈로 보내고 관리패킷인 경우 범주결정과 필터링을 하는 reply모듈로 보내게 된다. 이러한 모델의 각 모듈은 FSM(finite state machine)으로 기술되는데 reply 모듈의 경우 그림4와 같다.

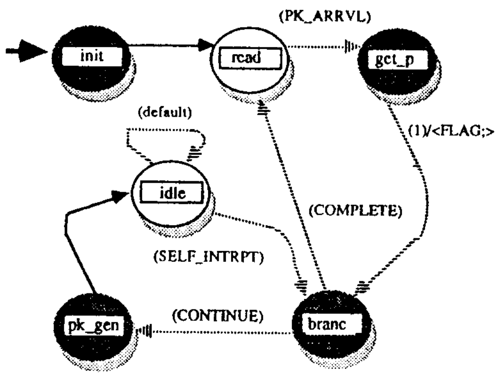

그림 4. reply 모듈의 내부처리구조
Fig. 4. Internal processing structure of reply module.

Reply 모듈은 관리요청패킷에 대한 응답을 발생시키기 위해 범주결정과 필터링을 행한다. 범주의 4가지 형태중 하나와 관련된 파라메터를 참조하여 응답을 위한 피관리객체를 선정한다. 유입되는 관리정보트리의 레벨값은 그 레벨에서 정해진 피관리 객체수를 결정할 수 있는 요소로 작용한다. 이때 관리정보트리의 수준 T_l에 대한 피관리객체의 수 $MO_l(T_l)$은 지수함수로 결정된다. 범주수준 3이나 4처럼 관리정보트리의 임의의 수준에 존재하는 관리객체의 수와 그 전의 수준에 존재하는 모든 관리객체를 원하는 경우 각 수준에서의 관리객체의 합 $MO_K(T_K)$로 나타난다.

$$MO_l(T_l) = e^{\alpha Tl} \ (\alpha \text{는 } 0.7\text{로 가정})$$

$$MO_K(T_K) = \sum_{l=0}^{K} MO_l(T_l)$$

Pk-gen 프로세스는 실제 응답 패킷을 발생시키고 idle 프로세스로 간다. idle 프로세스에서는 매 10ms 마다 인터럽트가 발생하도록 스케쥴링 하였으며 정해진 수의 응답패킷을 발생시키기 위해 branch 프로세스를 거쳐 다시 pk-gen 프로세스로 간다. 전송해야할 패킷을 모두 발생시키면 branch 프로세스에서 COMPLETION을 일으켜 ready 프로세스로 향하게 된다. 이때, 응답패킷은 필터링을 거친 패킷이 되는데 이 부분을 처리하는 프로세스는 get-pk 프로세스이다. 필터링은 균등확률 분포함수에 따라 발생하는 확률치 P_F를 이용한다. 필터링을 거친 관리응답패킷 수 $F(MO(T_{|l,K|}))$은 다음과 같다.

$$F(MO(T_{|l,K|})) = P_F{*}MO(T_{|l,K|})$$

위 식에서처럼 범주의 결과에 필터를 확률로 적용하여 최종응답 패킷수를 결정한다. Sink 모듈은 패킷을 파기시키는 프로세스 모델을 갖고 있다. Ethernet을 시뮬레이션 하기 위해 bus-rx, mac, defer, bus-tx 모듈이 존재하는데 이 모듈들은 기본적으로 제공되는 모델이다. 또, 피관리 시스템에 관한 시뮬레이션 모델은 m-gen 모듈을 제거하면 된다. 이 모델에서 전송매체에서의 지연시간 t_d는 수신 mac모듈에 도착한 임의의 패킷수신시간 t_r에서 이 패킷이 송신된 시간 t_s의 차로 알 수 있다. 또, 복수응답을 요구하는 범주형태에서는 이들의 평균시간이 지연시간이 된다.

$$t_d = \frac{\sum_{i=1}^{n} (t_r(i) - t_s(i))}{n} \quad (n\text{은 1보다 큰 응답 패킷수})$$

망에서 요구되는 부하 G는 망내에서 발생하는 패킷수 pkt(i)의 합을 시뮬레이션시간 t_{sim}으로 나눈것이 된다. 이때 G에 따라

$$G = \left(\sum_{i=1}^{n} pkt[i]\right) / t_{sim}$$

나타나는 요구응답시간 t_{res}는 요청패킷이 전달되는 시간 t_{req}, 내부 처리시간 t_{orec}, 각 응답 패킷의 지연

시간 D(i)의 합으로 나타난다.

$$t_{res} = t_{req} + t_{proc} + \sum_{i=1}^{n} D(i)$$

Ⅳ. 실험결과 및 분석

Ethernet에서 하나의 관리시스템이 다수의 피관리 시스템을 관리하는데 있어서 범주결정과 필터 파라미터의 특성검출, 그리고 그 결과 분석은 다음과 같다. 관리시스템이 주기적으로 30개의 피관리 시스템에게 문의하는 경우 그 시간에 따라 응답시간은 크게 변할 수 있다. 특히, 각 범주형태별로 살펴보는 것은 관리시스템의 구성과 관리프로토콜의 구현시 중요한 구현요소로 작용할 수 있다. 실제로 '관리시스템은 관리 정보의 무결성(integrity)을 지원하기 위해 폴링타임을 적절히 선택해야 하며 각 범주형태에 따른 폴링타임의 선택은 망의 트래픽 조절에 영향을 줄 수 있다. 폴링타임에 따른 요구응답 시간은 그림5와 같다.

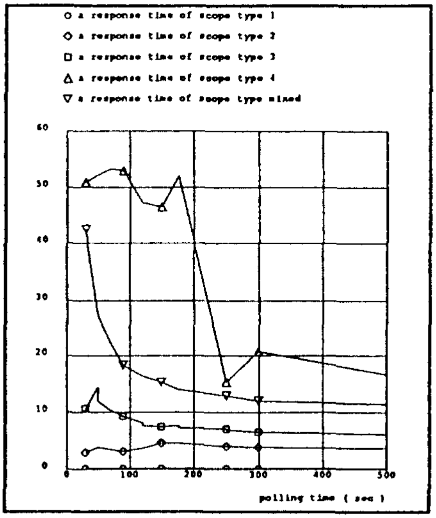

그림 5. 폴링타임에 따른 각 범주 형태별 요구응답시간

Fig. 5. A response time of each scope type according to a polling time.

범주형태 1과 2에서 요구응답시간은 폴링 타임의 값에 영향을 거의 받지 않는다. 범주형태 3에서의 요구응답시간은 폴링타임이 100sec 일때를 기준으로 안정화되는 추세를 보이며 범주형태 4는 250sec를 경과해서 점차 안정화 되는 것을 보여주고 있다. 범주형태를 혼용하는 경우는 300sec를 지나서는 큰 변동없이 안정화되나 100sec 미만인 경우와 그 이후는 큰 편차를 보여주고 있다. 범주형태에 따라 나타나는 요구응답시간은 피관리 시스템으로 부터 얼마나 많은 양의 정보를 가져오는가에 의존적이다. 따라서 피관리 시스템으로부터 많은 양의 정보를 획득하기 위해서는 범주형태 4가 가장 이상적이나 이 형태만을 이용하는 경우, 그림5에서와 같이 빈번한 폴링은 요구응답시간면에서 부적절함을 알 수 있다. 요구응답시간의 개선을 위하여 폴링시간을 증가시키면 관리시스템은 피관리 시스템의 상태를 정확히 파악할 수 없으며 이는 무결성의 문제를 야기시킨다. 이와 같이 적절한 요구응답시간의 요구사항과 피관리 시스템으로 부터 원하는 양의 정보를 조회하는 경우 가장 적합한 형태는 모든 범주형태를 이용자가 필요에 따라 변경하여 사용하는 것임을 알 수 있다. 이때 적용할 각 범주형태별 최적 폴링시간의 값은 표3과 같다.

표 3. 범주형태와 적정 폴링타임

Table 3. A scope type and related polling time.

scope type	polling time
type 1	1 sec ≧
type 2	10 sec ≧
type 3	100 sec ≧
type 4	250 sec ≧

전송매체에서 각 응답패킷에 대한 지연시간은 망에 인가되는 패킷의 영향을 받게 된다. 폴링타임을 120sec로 일반패킷발생율 $\lambda = 2$(packets per second)로 고정시키고서 부하에 따른 각 범주 형태별 지연시간은 그림6과 같다.

그림6에서 나타내듯이 범주형태 1과 2는 관리트래픽 발생자체가 미미하여 지연시간 측면에서 큰 영향을 주고있지 않고, 범주형태 3은 망의 부하에 따라 선형적으로 증가하다가 부하가 70정도에서 지연시간이 크게 상승하고 있다. 범주형태 4는 자체가 발생시키는 응답 패킷수가 망 부하에 영향을 크게 주어 패킷전달 지연시간은 점차 증가하게 되고 부하가 240정도에서는 지연시간을 급격히 증가시킨다. 이와 같은 현상은 자체에서 발생시키는 부하가 망에 미치는 영향이 크게 되고 패킷을 모두 전달하기 전에 또 다른 관리요청 패킷의 수신으로 점차 대기행렬에 대기하는 시간이 늘어나고 일정부하 이상이 되면 전달시간에 비해 대기행렬에 쌓이는 속도가 빨라서 지연시간은 폭주하게 되는 것이다. 그러므로, 망의 부하

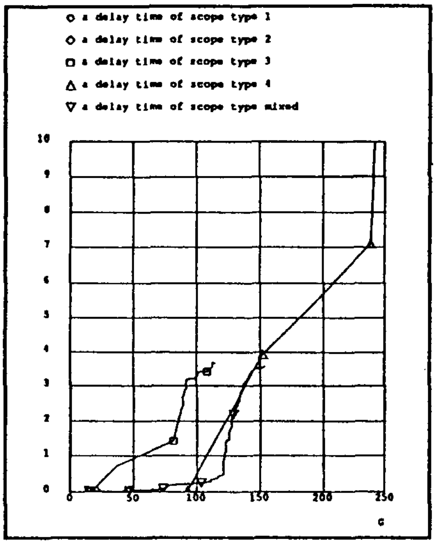

그림 6. 망의 부하에 따른 각 범주 형태별 전송매체지연시간

Fig. 6. A delay time of each scope type according to an offered load.

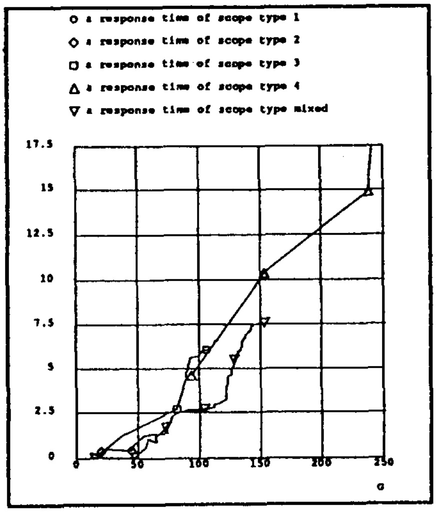

그림 7. 망의 부하에 따른 각 범주형태별 요구응답시간

Fig. 7. A response time of each scope type according to an offered load.

에 따른 지연시간 면에서 범주형태 4를 이용하는 것이 가장 비현실적이고 모든 범주 형태를 이용하는 것이 전반적으로 좋은것을 알 수 있다. 망의 부하에 따른 요구응답시간은 관리 시스템이 피관리 시스템에게 관리 패킷을 보낸 후 얼마만큼의 시간 경과 후 그 응답을 받을 수 있는가에 대한 문제이다.

범주형태 1과 2는 요구응답시간면에서 짧은 시간을 요구하나 피관리객체의 범주 설정상 다양한 능력을 제공하지 못하여 비현실적이며 범주형태 3은 망의 부하에 민감한 것으로 나타났다. 범주형태 4는 자체가 제공하는 망의 부하로 인해 응답시간이 큰폭으로 상승하여 망에 따른 요구응답시간 면에서 가장 비현실적이다. 이러한 현상은 시스템 대기행렬과 밀접한 관계를 갖는데, 피관리 시스템의 처리능력을 넘는 응답발생과 망의 부하가 관리패킷의 대기행렬 형태를 발생하여 응답시간이 지연됨에 따라 그 뒤의 관리 요청에 대한 응답은 점점더 대기행렬에서 대기하게 된다. 반면 여러범주형태를 동시에 이용하는 경우는 망의 부하에도 덜 민감하고 peak요구 응답시간면에서나 피관리객체 선정의 융통성면에서 좋은것으로 나타났다. 특히, 부하가 80에서 120사이에서 안정된 형태를 보이는데, 이는 망의 부하가 커진다 하여도 여러형태의 범주를 이용함에 따라 빠른 응답을 받는 경우가 발생하기 때문이다. 즉, 나중에 요청한 관리패킷이 먼저 응답을 받는 경우가 이 범주형태에서는 자주 발생한다. 그러나, 이것도 일정 망의 부하를 넘어서는 경우 응답시간이 급격히 상승하게 된다. 이렇게 요구응답시간과 지연시간은 대기행렬의 형태와 아주 밀접하다는 것을 알 수 있다. 그래서 그림 8에서 망의 부하에 따른 대기행렬용량 추이를 관찰하였다. 범주형태 1과 2는 대기행렬용량이 1을 넘지 못하여 그림에서 삭제되었다. 범주형태 3에서는 망부하가 40이전까지는 대기행렬이 일어나지 않다가 그 시점에서 상승하게 된다. 또, 100을 지남에 따라 상승곡선을 이룬다. 범주형태 4는 항상 일정 망 부하 90이상을 갖으므로 그 이하에서는 트래픽이 없어서 대기행렬이 없는 것으로 나타났다.

이와같이 범주형태 3인 경우가 효율적임을 알 수 있으며 대기행렬용량과 요구응답 시간과의 관계를 보면 그림9와 같다.

대기행렬용량과 요구응답시간과의 관계를 보면 범주형태 4에서 요구응답시간의 개선을 위해 범주용량을 늘리는 것은 큰 의미가 없다. 즉, 대기행렬용량이 증가하여도 요구응답시간이 감소하는 것이 아니라 비례하여 증가한다. 이는 응답패킷수가 급증함에 따라 기본적으로 필요한 대기행렬용량이 증가하며, 대기행렬용량의 제한은 패킷의 손실을 의미하게 된다. 그러므로 응답패킷의 증가는 대기행렬용량을 증가시키고 요구응답시간도 증가시키게 된다. 반면 범주형태 3과 혼용하는 형태는 대기행렬의 용량이 증가하

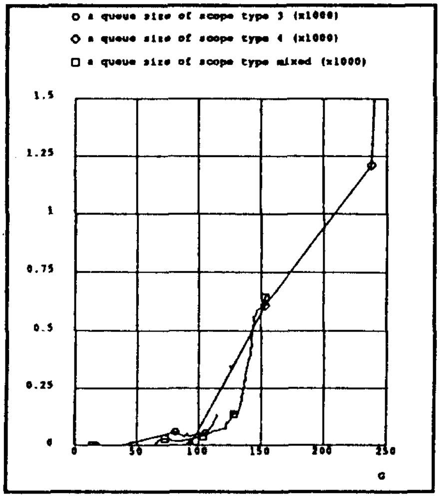

그림 8. 망의 부하에 따른 각 범주 형태별 대기행렬용량

Fig. 8. A queue size of each scope type according to an offered load.

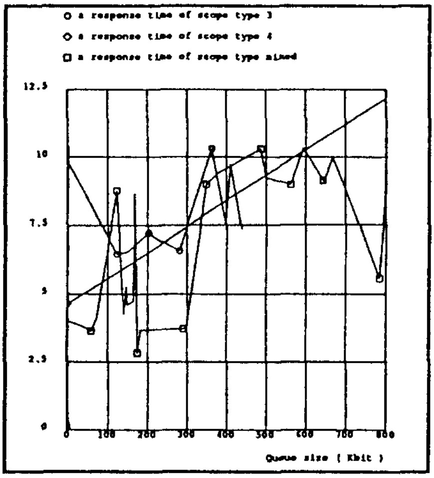

그림 9. 대기행렬용량에 따른 각 범주 형태별 요구응답시간

Fig. 9. A response time of each scope type according to queue size.

여도 응답시간이 감소하는 것이 아니라 어느정도 간격을 두고 증감한다. 이는 망의 부하가 증가하면 대기행렬로 유입되는 패킷은 감소하고 자체 서비스율 μ에 의해 소모되므로 대기행렬용량이 감소하고 응답시간은 증가한다. 반면 망의 부하가 순간적으로 적은 경우 대기행렬에 유입되는 응답패킷이 증가하여 대기행렬용량은 증가하고 응답이 망을 경유해 빠르게 도착하므로 응답시간은 감소하게 된다. 이와같이 대기행렬용량과 요구응답시간 사이의 분석은 범주형태 3과 혼용하는 경우가 비교적 안정적임을 알 수 있다.

V. 결 론

본 논문에서는 표준 망 관리 프로토콜에서 범주결정/필터 파라미터를 대상으로 시뮬레이션 모델을 정하고 성능분석을 하여 망관리 시스템에서의 최적 폴링시간을 제시하고 망의 부하에 따른 대기행렬용량, 요구응답시간, 전송매체 지연시간을 각 범주형태에 따라, 그리고 범주형태를 혼용하는 경우로 나누어 비교 분석하여 적정 범주형태를 제안하였다. 망관리 시스템에서 현실에 맞는 응답시간을 얻기 위해서는 폴링타임의 선택이 중요한 문제가 된다. 폴링타임을 증가시키면서 응답시간을 측정한 결과 최적의 폴링타임이 나타났으며 망관리시스템에서 범주형태를 혼용하여 사용하는 경우는 200sec로 나타났다. 특히, 범주형태 4만을 이용하는 망은 비현실적이며 여러 범주형태를 동시에 사용하는 망관리시스템은 응답시간적으로 범주형태 3과 비슷한 성능을 얻게 된다. 망의 부하를 변경시키면서 지연시간, 응답시간, 대기행렬용량을 측정하였는데 이 경우 범주형태 4는 모든 면에서 바람직하지 않다는 결론을 얻었으며 각 범주형태는 망의 부하에 민감하게 반응하는 것을 알 수 있었다. 또 대기행렬용량과 응답시간의 분석 측면에서 보면 범주형태를 혼용하는 경우가 가장 성능이 좋음을 알 수 있다. 피관리 객체 선정에 있어서 조건은 필터를 이용하게 되는데, 실제로 일어나는 상황은 가변적이므로 균등확률 분포함수를 이용하여 이들 성능 지수들을 측정하였다.

결론적으로 망에서 시스템의 능력과 망의 부하를 고려하여 가장 적합한 범주형태는 모든 범주형태를 조합하여 사용하는 것이며, 망관리시스템은 이들의 사용시 본 논문에서 제안한 성능지수를 고려하여 각 범주형태에 적용하는 것이 성능향상을 이룰 수 있다.

參 考 文 獻

[1] S. Mark Klerer, "The OSI management architecture: an overview," *IEEE Net.*, Megazine, vol. 2, no. 2, pp. 20~29.

[2] Jock Embry, Peter Manson, Dave Milham, "An open network management architecture: OSI/NM forum architecture and concepts," *IEEE Net.* Magazine, pp. 44~53, July 1990.

[3] M. Feridun, M. Leib, M. Nodine, J. Ong, "ANM:automated network management system," *IEEE Net.* Magazine, vol. 2, no. 2, pp. 13-19, March 1988.

[4] Douglas E. Comer, "Internetworking with TCP/IP(vol. I)," Prentice Hall. pp. 403~420, 1991.

[5] Information Processing Systems-Open Systems Interconnection, "System management overview," ISO/IEC 10040, 1990.

[6] Information Processing Systems-Open Systems Interconnection, "Basic reference model-management framwork," ISO/IEC 7498-4, 1989.

[7] Information Processing Systems-Open Systems Interconnection, "Common management informations service definition," ISO/IEC 9596, May 991.

[8] Information Processing Systems-Open Systems Interconnection, "Common management information protocol specification," ISO/IEC 9595, May 1991.

[9] L.N. Cassel, C. Patridege, J. Westcott, "Network management architectures and protocols: problems and approaches," *IEEE Selected Areas in Comm.*, vol. 7, no. 7, pp. 1104-1113, sep. 1989.

[10] Marshall T. Rose, "The Simple Book," Prentice Hall, pp 91~130, 1991

[11] K. McCloghrie, "Management information base for network management of TCP/IP-based internets," RFC 1156, May 1990.

[12] Ian Sugarvroad, "An OSI-based interoperability architecture for managing hybrid Networks," *IEEE Net.* Magazine, pp 61~69, March 1990.

[13] Fred Halsall, Nasser Modiri, "An OSI-based interoperability architecture for managing hybrid networks," *IEEE Net.* Magazine, pp. 61~69, March 1990.

[14] W.Bux, "Local area Subnetworks: a performance comparison," *IEEE Trans. on Comm.*, vol. COM -29, no. 10. pp. 1465-1473, Oct. 1981.

[15] Information Processing System-Open Systems Interconnections, 'Basic reference model." ISO/IEC 7498~4

[16] Alain C. Jerome, Steven P. Baraniuk, "Communication network and protocol design automation," *IEEE Military Comm.* Conference, pp .334~339, Oct. 1987.

[17] 박희동, 임효택 외2, """OSI 관리프로토콜에서 관리정보범주 결정방안의 분석,"동계컴퓨터 통신워크샵 논문집, pp. 217~224, 1991년 1월.

[18] 안성진 변옥환 외3, "Internet상에서 공통관리정보서비스와 프로토콜(CMOT)의 설계 및 구현," 동계컴퓨터통신워크샵 논문집, pp. 192~201, 1992년 1월.

著 者 紹 介

邊 玉 煥(正會員)
1953年 8月 28日生. 1979年 2月 한국항공대학 통신공학과(공학사) 1985年 인하대학교 대학원 전자공학과(공학석사). 1988年 3月~현재 경희대학교 대학원 전자공학과(박사과정수료). 1978年 9月~현재 KIST 시스템공학연구소 교육연구망 실장. 1983年 12月~1984年 12月 미국 OMS Computer corp. 연구원. 주관심분야는 Network Management, Network Security 등임.

陳 庸 玉 (正會員) 第29卷 A編 第3號 參照
현재 경희대학교 전자공학과 교수