

論文93-30A-3-1

## OSI 망관리 시스템에서 사건보고 관리를 위한 1진 피드백 과보고 회피기법

### (The Unary Feedback Over-Reporting Avoidance Scheme for the Event Report Management on the OSI Network Management System)

邊 玉 煥\*, 陳 庸 玉\*\*

(Ok Hwan Byeon and Yong Ohk Chin)

#### 要 約

본 논문에서는 피관리시스템에서 관리시스템으로 사건보고를 하는 OSI망관리모델에서 피관리 시스템의 과보고를 조절하여망 관리 트래픽의 혼잡을 회피하는 과보고 회피 기법을 제안한다. 피관리 시스템에서 관리시스템으로 사건보고를 하는 경우 관리 트래픽 집중현상이 발생하여 관리시스템에게 과부하를 줄 수 있으며 네트워크에 혼잡을 발생시킬 수 있다. 이 기법은 관리 시스템에서 피관리 시스템으로의 피드백을 이용하게 된다. 피관리 시스템은 자신에게 설정된 단위시간당 최대수 만큼의 사건보고를 관리 시스템에게 전송하며, ThresholdTime에 이르면 자신의 관리변수를 LOCK 상태로 변경하고 사건보고를 중지한다. 이때 관리시스템은 자신의 상태를 참조하여 피관리 시스템에게 M-SET 프리미티브를 이용하여 사건보고 재개를 지시한다. 이 기법은 분산 처리가 가능하고, 망의 동적인 상황에 잘 적응하며 최적의 운영점으로 수렴하고, 공정성을 보장한다. 1진 피드백 과보고 회피기법의 특성을 검출하기위해 ThresholdTime값을 측정하여 피관리 시스템의 사건보고 조절능력과 각 노드에서의 공정성 보장을 관찰하였다. 평균 활성화 노드수와LOCK 지속시간을 사건보고 부하에 따라 측정하고 일반패킷 부하의 영향을 관측하기 위해 패킷 분실율, 관리자 큐 지연시간, goodput을 측정하였다. 그리고, 이진피드백기법, OSI 망관리 시스템의 순수기법과 시뮬레이션을 통해 상호 비교 분석하여 본 논문에서 제안한 기법의 우수성을 입증하였다.

#### Abstract

In this paper, we propose over-reporting avoidance scheme which avoids congestion of network traffics by adjusting managed system's over-reporting, on the OSI network management model which reports events from managed system to managing system. In case of reporting events from managed system to managing system, management traffic concentration occurs, and it causes over-loading on the managing system and congestion on the network. This scheme takes advantage of feedback from managing system to managed system. Managed system transmits event reports as much as maximum event pertime allocated to itself to managing system, and it sets it's management variables to LOCK state and stops event reports as Threshold time is reached. At the time, managing system directs event reports again by using M-set primitive with referring it's status. With this scheme, distributed processing, dynamic network adaptation, convergence of optimal operation point is possible. In addition to it, a fairness is assured. In order to detect characteristics of the Unary feedback over-reporting avoidance scheme, It is observed a control capability of the event reporting and fairness of each nodes through measuring ThresholdTime value. It is measured a number of mean activating nodes and maintained time of LOCK state according to event reporting load, and also measured lost ratio of management packet, queuing delay in managing system, and goodput to observe effects of general packet load. Binary feedback scheme, Unary feedback over-reporting avoidance scheme and raw scheme on the OSI network management system each are compared and analyzed, and finally proved that the scheme proposed in this study performs better.

\*正會員, 시스템工學研究所  
(System Engineering 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 10月 29日

## I. 서론

최근 컴퓨터와 통신기술의 급속한 발달 및 시스템의 통합화 추세와 더불어 전산망을 사용하고자하는 이용자들의 급증에따라 전산망내의 정보통신 양이 급격히 증가하고 있다. 이에따라 정보를 차질없이 전달 해주어야하는 전산망을 안전하면서도 효율적으로 운영 하기위해 정보통신 서비스를 제공하는 통신자원을 어떻게 관리하느냐가 매우 중요한 문제로 부상하게 되었다.

이러한 전산망 관리의 표준화를 위해 현재 ISO에서는 통신장비나 관리형태와 무관하게 전체 망을 통합된 망관리 시스템을 이용하여 관리 할 수 있도록 OSI 관리 표준을 제안하고 있다. OSI에서의 시스템 관리 표준은 기본구조표준, 관리통신표준, 관리대상 표준, 시스템관리기능표준으로 대별하여 표준화 작업을 진행하고 있으며 이에 근거한 표준망관리시스템 제품들이 선을 보이기 시작하고 있다.<sup>[1, 2, 3]</sup>

한편 전산망 관리시스템에서 관리행위의 수행은 기존 트래픽에 최소한의 영향을 주면서 최대의 관리기능을 수행해야하는 관리목표와 문제발생시 가장 빠른 시간내에 문제해결을 해야하는 서로 상반되는 여건을 어떻게 잘 조화시켜 관리를 수행하느냐가 관리의 잇점을 최대로살리는데 있어서 중요하다. 모든 관리 트래픽의 교환은 망의 성능을 저하시키는 요인이 되므로 관리 트래픽을 효율적으로 제어 할수 있는 방안이 모색되어야 할 것이다.

특히 관리시스템이 자의적으로 트래픽을 조절할수 없는 보고 서비스의 경우에 있어서는 망의 성능에 많은 영향을 미칠것이다.

본 논문에서는 피관리시스템에서 관리시스템으로 자기에 속한 관리 객체에서 방출되는 사건보고를 하는 망관리 모델에서 피관리시스템의 과보고를 조절하여 망관리 트래픽의 혼잡을 막아줄 1진 피드백 과보고 회피 기법을 제안하고자 한다. 망관리 트래픽제어와 관련된 연구는 관리 시스템에서 피관리 시스템으로 발하는 운영 절차중 SCOPE/FILTER, M-CANCEL-GET등의 특성을 성능분석하여 적정 방안을 제시한 일부 연구가 있으나<sup>[18, 19]</sup>, OSI망관리 환경에서 관리 시스템의 부하를 줄여주고, 망관리 트래픽의 많은 부분을 차지하는 Event Report 관련 트래픽을 효율적으로 제어 해주는 기법에 대한 연구활동은 보기가 힘들다.<sup>[8]</sup> 단지 네트워크 계층을 중심으로한 일반 패킷의 흐름제어나 혼잡방지의 효율적 제어를 위한 Binary Feedback Scheme등의 주요 연구는 많이 이루어지고 있다.<sup>[9, 15]</sup>

본 연구에서는 Mil3 Inc. 의 시뮬레이션 도구인 OPNET을 이용하여 Ethernet상에서 사건보고처리와 관련된 관리시스템과 피관리 시스템의 시뮬레이션 모델을 정하고 Threshold 시간의 변화, LOCK상태 지속시간, LOCK상태에 있는 시스템수, 관리시스템의 Queuing Delay, 관리 패킷 분실 확률등을 분석하여 본 논문에서 제안한 기법의 특성을 추출하고 그 우수성을 입증하고자 한다.

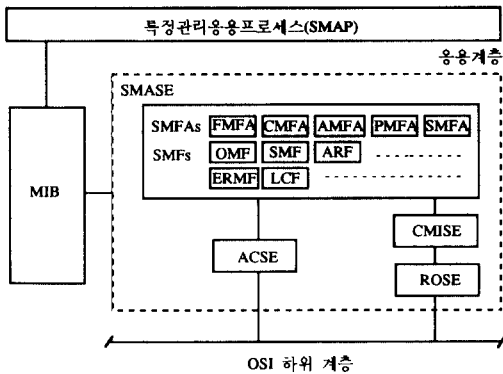
2장에서 사건보고 관리 기능 및 동작요청, 3장에서는 제안되고있는 1진 피드백 과보고 회피기법의 상세 설명, 4장에서 시뮬레이션 모델설정, 5장에서는 실험의 결과를 비교분석하고 마지막으로 본 논문의 결론 및 앞으로의 연구방향에 대하여 논한다.

## II. 사건보고 관리기능 및 동작

망 관리는 관리대상의 집합체, 관리 프로토콜, 관리 응용 서비스등으로 구분되어 표준화 되고 있다.<sup>[1]</sup> 관리대상 [Managed Object]의 집합체는 관리정보 베이스인 MIB라는 논리적인 집합체로 구성되며 관리 프로토콜인 CMIP [Common Management Information Protocol]은 관리응용서비스의 메시지 교환을 담당한다.<sup>[2, 3, 17]</sup> 마지막으로 관리응용 서비스 요소인 SMASE [Systems Management Application Service Element]는 관리행위를 위한 서비스를 제공하기 위해 존재한다. 이 중에서 시스템 관리 응용 서비스요소는 그림 1과 같이 5개의 시스템 관리 기능영역 [SMFA]과 다수의 시스템관리기능 [SMF]으로 나뉘어져 작용한다.<sup>[1, 16]</sup> 예로서 고장 관리 기능영역 [FMFA]은 고장관리를 위한 에러기록의 유지와 검사, 에러발견 통보와 이에따른 활동, 고장 상태추적, 연속적 진단 시험수행, 고장상태의 교정 등을 수행하는 기능등을 포함한다.

시스템 관리 응용 서비스 요소는 관리 시스템과 피관리 시스템 모두에 존재하며 메시지의 교환을 위하여 CMISE에게 관리운영, 관리통고 방식으로 서비스를 요청한다. 이때 시스템관리기능중 하나인 사건보고 관리기능 [ERMF]은 관리자에게 관리 객체에서 방출하는 사건보고들의 전송을 제어 할수 있게 해준다.<sup>[6]</sup> 이를 수행하기 위해서 사건보고 관리기능은 사건발송 판별자 [Event Forwarding Discriminator]라는 하나의 관리 지원 객체를 정의한다. 사건 발송 판별자는 특정관리객체 또는 관리객체 집합에 관한 잠재사건보고를 명시된 목적지로 보낼것인가 보내지 않을 것인가를 결정하는 관리지원객체이다. 잠재사건보고는 관리객체에서 방출된 통고

[notification] 들로부터 유도된 정보와 그 통고를 처리하는 과정에서 파생된 정보들로 구성되는 개념적 객체를 의미한다. 결국, 사건보고 전송의 제어는 피관리시스템내에 정의된 사건보고판별자 관리지원객체에 대한 관리오퍼레이션에 의해 이루어진다. 즉, 사건보고판별자 관리지원객체의 생성과 삭제 오퍼레이션으로서 사건보고 발생의 시작과 종료를 제어할 수 있으며, 사건보고판별자가 갖는 관리상태에 대한 속성값을 변경하는 오퍼레이션으로서 사건보고 발송의 일시중지와 재시작을 제어할 수 있게 된다.



- FMFA(Fault Management Functional Area)
- CMFA(Configuration Management Functional Area)
- AMFA(Account Management Functional Area)
- PMFA(Performance Management Functional Area)
- SMFA(Security Management Functional Area)
- OMF(Object Management Function)
- ERMF(Event Report Management Function)
- SMF(State Management Function)
- ARF(Alarm Reporting Function)
- LCF(Log Control Function)

그림 1. 시스템 관리 응용서비스 요소의 구성  
Fig. 1. Configuration of systems management application service element.

또한 사건보고판별자는 사건보고의 발생여부를 결정하기위한 검사기준을 명시하는 판별자 구문이라는 속성을 갖는데, 이 속성의 값을 조정함으로써 특정 사건보고에 대한 발송여부를 제어할 수 있게 된다.

그림 2는 사건의 생성, 처리, 보고에 관계된 요소들에 대한 도식적인 설명을 보이고 있다. 사건보고 판별자 역시 관리객체이어서 사실상, 사건보고 판별자에서 방출되는 통고들도 다른 모든 통고들과 마찬가지로 관리정보베이스로부터 발생하는 것이다.

사건보고 관리기능은 사건보고 전송의 제어를 위해 사건보고 판별자라는 일종의 관리 객체를 정의하고

그것에 대한 관리 오퍼레이션을 통하여 원하는 사건 보고 관리를 행한다.

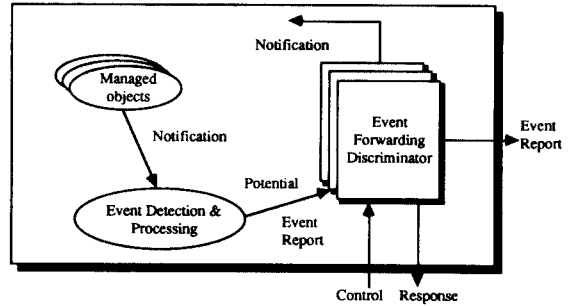


그림 2. 사건보고 관리 모델  
Fig. 2. Event reporting management model.

사건보고 관리자가 피관리자에게 사건보고 판별자를 생성하도록 요구할 때에는 M-CREATE 서비스를 이용하며 삭제하도록 요구할 때에는 M-DELETE 서비스를 이용한다. 이와 유사하게 사건보고 판별자의 속성값을 변경하기 위해서는 M-SET 서비스를 이용한다. 사건보고관리 기능은 이러한 관리활동을 이루기 위해서 객체관리기능 [OMF] 에 의해 제공되는 통과서비스<sup>1)</sup>, 로그제어기능 [LCF] 에 의해 제공되는 로깅서비스 그리고 상태관리 [SMF] 기능 등을 이용하며<sup>15)</sup> 이들 여러기능과 네트워크에서 사용하는 일반적인 혼잡방지기법의 개념을 도입, 연계시켜 사건보고시 트래픽을 통제하는 방안을 기대할 수 있다.

이들 중에서 로그 제어 기능은 관리자로부터 하여금 관리를 위해 필요한 정보의 로깅을 제어 할 수 있도록 해준다.<sup>16)</sup> 여러가지 관리기능을 위해서, 발생되었던 사건들에 대한 정보와 여러 관리객체들에 대해 또는 그 관리객체에 의해서 발생하는 관리 오퍼레이션들에 대한 정보를 보유할 필요가 있다. 실제의 개방형 시스템에서는, 이러한 정보를 저장하기 위해서 여러가지 자원들이 할당될 수 있다.

### III. 1진 피드백 과보고 회피 기법

네트워크 관리에서 사용되는 사건보고 관리는 각 피관리 시스템에서 발생하는 공지사항을 관리시스템에게 통보하는 서비스를 제공한다. 이때 공지사항은 피관리객체의 오류, 경고등을 포함하게 되며 경보 보고 기능 [alarm reporting function]으로 표현된다. 이와같이 사건보고는 다수의 피관리 객체가 관리시스템으로 집중하는 성향을 갖게되는데 이는 네트워크의 게이트웨이에서 발생하는 혼잡 [congestion] 문제

와 동일한 현상이 발생할수있다. 즉, 서브네트에 존재하는 여러 시스템이 다른 서브네트로 패킷을 전달하고자 하는 경우 게이트웨이를 통과해야 함으로 게이트웨이에 트래픽이 집중하여 혼잡을 유발할 수 있는것이다.<sup>12, 14, 15</sup> 망관리에서도 이와 유사하게 관리객체의 사건보고가 관리시스템으로 집중하는 특성을 갖고 있음으로 인해 과보고가 발생할 수 있으며 이로 인해 혼잡을 유발할 수 있다.<sup>6</sup>

본 논문에서는 이러한 과보고현상을 회피하는 기법으로 1진 피드백 과보고 회피기법을 제안한다. 과보고 회피기법은 과보고 현상이 일어나지 않도록 하기 위하여 관리시스템으로 유입되는 사건보고를 조절함으로써 이루어진다.

사건보고의 유입양은 피관리 시스템의 Threshold-Time과 관리시스템으로부터의 피드백 정보를 이용함으로써 이러한 기법을 본 논문에서는 Threshold메카니즘을 갖는 1진 피드백 과보고 회피기법이라 하였다. 여기에서 피드백의 개념은 네트워크 시스템에서의 Binary feedback congestion avoidance 기법<sup>9, 10</sup> 및 사건보고판별자에서 제공하는 데이터제어 개념<sup>16</sup>을 참조하여 관리시스템에서 LOCK 해제신호만을 피드백 시키는 개념으로하여 피관리시스템과 관리시스템이 관리트래픽조절에 대해 공동 책임을 갖는 특성을 갖는 Unary Feedback 개념이다. 이 기법이 망관리와 서로 연관되어 동작하기 위해서는 표 1과 같은 관리변수의 정의가 필요하다.

administrativeState는 사건발송 판별자의 운영 상태를 나타내는 것으로 LOCK과 UNLOCK의 2가지 값을 갖는다. 이 값은 사건보고의 허가여부를 나타내기 위해 이용된다.

EventReportDisabledQ는 피관리 시스템으로부터 운영상태변경, 즉 사건보고가 금지되었다는 보고를 받을때, 관리시스템 자신의 사건보고 처리상태가 좋지 않은경우, 곧바로 피드백 (피관리 시스템에게 사건보고 재개를 알리는 행위)을 할수없게 되는데, 이때 사건보고가 금지된 피관리 시스템이 어느것이 보고된 순서가 어떤지를 후에 알수있도록 이 큐에 기록한다. maxEventPerTime은 단위시간당 발생할수 있는 사건보고 수를 나타내는 상수이다. TimeQ는 maxEventPerTime만큼 존재하며, 최근의 maxEventPerTime개의 사건보고 발생시간에 대한 기록을 유지한다. 피관리 시스템이 자신의 사건보고율을 계산하기위해 각각의 사건 발생시간을 기록한후 일정수의 사건보고가 일어난 시간간격을 계산함으로써 자신의 사건보고율이 높은지낮은지를 알 수 있게 된다. 이때 시간간격을 조절하는 변수가 Threshold-Time

이다. 이러한 변수들은 망관리 시스템 내부에서 서로 연관되어 동작하며 이들의 변화, 특히 운영상태의 변화는 administrativeState상태변경통지와 상태변경보고 서비스라는 망관리 서비스를 이용하게 된다.

administrativeState상태변경통지는 사건발송 판별자의 운영상태가 UNLOCK에서 LOCK으로 또는 LOCK에서 UNLOCK으로 변경된 사실을 알리는 통지서비스이다. 상태변경보고 서비스는 피관리시스템에서 방출된 상태변경통지를 관리시스템에게 보고하기위해 피관리 시스템이 사용하는 서비스이다. 피관리 시스템이 자신의 사건보고가 금지되었음을 관리자에게 알리기 위해서 사용되며, 이보고는 확실하게 관리시스템에 보고되어야 하므로 확인형 서비스가 사용된다. 이보고가 분실되는 경우, 관리시스템이 피드백을 할수없게 되므로, 피관리자는 Deadlock상태에 이르게 된다.

표 1. 관리변수정의

Table 1. Definition of management varizbles.

관리 변수	의 미
administrative State	사건발송판별자의 속성 (attribute)
EventReportCisabledQ	administrativeState가 LOCK인 피관리시스템의 응용 엔티티 타이틀을 갖는 큐
maxEventPerTime	사건보고율을 계산하기 위해 필요한 상수로 단위사건 보고수를 나타낸다.
TimeQ	각 사건 보고 발생시간을 기록하는 환형큐
ThresholdTime	계산된 시간차이를 비교하는 기준시간

관리시스템이 상태변경보고를 받게되면 자신의 큐를 검사하여 상태가 좋은경우 바로 M-SFT를 이용하여 피드백 시키고 상태가 좋지 않은경우는 큐상태가 회복된후 피드백을 시키게 된다. 큐의 상태는 H,L이라는 2가지 임계치를 이용하여 판단한다. 현재의 큐용량이 H를 넘어서는 순간부터 자신의 처리능력이 좋지 않다고 판단하며, L을 내려가는 순간 다시 회복되었다고 간주한다.

그림 3에서와 같이 사건보고가 유입되는 I, II 상태에서 LOCK이 발생하면 관리시스템의 큐용량이 바로 감소하고 UNLOCK이 발생하면 지연시간  $\tau_d$ 가 경과후 다시 관리시스템의 큐가 증가한다. 임계치 H를 경과후 LOCK이 발생하면 L이하로 큐용량이 감소할때까지 UNLOCK을 발생시키지 않는다. 이와같은 관리시스템의 UNLOCK 메카니즘을 가상코드로 보면 다음 그림 4와 같다.

피관리시스템은 운영상태를 UNLOCK으로 초기화하고 TimeQ를 초기화한다. 사건이 발생하면 자신의 사건발송 판별자의 운영상태를 조사하여 UNLOCK

이때 TimeQ에 현재시간을 기록하고 M-Event-Report 서비스를 이용하여 관리시스템에게 보고한다. LOCK상태라면 사건보고가 금지되었으므로 그 사건을 자신의 로그에 저장한다. 이때 로그가 포화상태가 될 수 있는데 이경우 가장 오래된 사건정보를 삭제하고 새로운 사건 정보를 저장한다. max-EventPerTime이 발생후 현재의 사건Te(K)와 TimeQ에 저장된 가장 오래된 사건보고 시간Te(i)와의 차이를 나타내는 Di,k를 다음 식과 같이 구한다.

$$D_{i,k} = T_c(k) - T_c(i)$$

Di,k가 ThresholdTime보다 작을경우, 피관리 시스템은 자신의 사건발송 관별자의 운영상태를 UNLOCK에서 LOCK으로 변경하고, 관리자에게 자신의 사건보고 금지상태를 보고한다. 이때, 상태관리기능이 제공하는 확인형 상태 변경보고 서비스를 이용한다. 관리시스템으로 부터 사건보고재개가 즉시 오면 피관리시스템의 운영상태를 LOCK에서 UNLOCK으로 변경하고 ThresholdTime을 감소시킨다. 반면 사건보고 재개가 즉시 수신되지 않는 경우 ThresholdTime을 리세트 시킨다. 사건 보고 금지가 해제된후, 로그에 저장되었던 사건들은 다시 관리 시스템으로 보고된다.

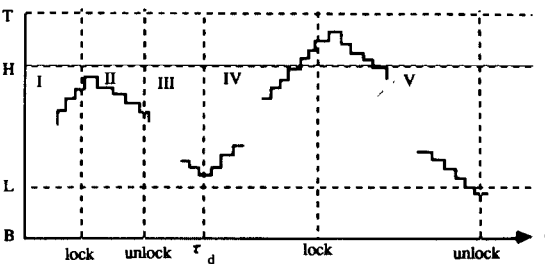


그림 3. 관리시스템의 큐상태  
Fig. 3. Queue States of management system.

그리고 ThresholdTime이 충분히 작아져 있다면 Di,k를 아무리 계산해도 ThresholdTime보다 작지 않을 것이다. 즉 이러한 경우 피관리 시스템은 LOCK상태로 되는 기회를 얻지 못하는데, 문제는 ThresholdTime이 충분히 낮아졌다는것은 피관리시스템의 요구량이 많았고 그 요구에 대해서 관리 시스템이 허가를 했다는것을 의미하기 때문이다. 이경우 피관리시스템이 계속 보고를 한다면 관리시스템의 일반 패킷 증가등으로 상태가 나빠져도 피관리시스템은

인식할 방법이 없다. 이러한 경우를 대비해서 Di,k를 ThresholdTime과 일정 횟수동안 비교하여 한번도 작지 않으면 관리 시스템의 상태만을 물어보는 기능을 수행한다. 이에 대한 관리시스템의 응답은 역시 GOOD 과 BAD이며 GOOD인 경우 피관리자는 아무런 일도 행하지 아니하고 원래의 조건대로 수행하며 BAD인 경우는 ThresholdTime값을 리세트시켜 다음 Di,k의 비교시 바로 LOCK상태로 변화도록하여 조절 기능을 행한다.

```

if (interrupted_signal == State_Change_Report.req) {
    if (H > Q_crit)
        Q_stat = GOOD ;
    else
        Q_stat = BAD ;
        LOCK = TRUE ;
    }
    send State_Change_Report.res (Q_stat) ;
    if (Q_stat == GOOD && LOCK == FALSE)
        send M_Set (UNLOCK)
    else
        insert the AE-title to EventReportDisabledQ ;
}
else if (interrupted_signal == CheckQ) {
    check the Queue ;
    if (L >= Q_crit) {
        dequeue the EventReportDisabledQ ;
        send M-Set (UNLOCK) ;
        LOCK = FALSE ;
    }
}
    
```

그림 4. 관리시스템의 UNLOCK알고리즘  
Fig. 4. UNLOCK algorithm of a management system.

이와같이 ThresholdTime은 피관리시스템의 LOCK메카니즘에 중요한 역할을 하게된다. 이 시간이 관리시스템의 상황에 맞지않는 경우 관리 시스템의 큐는 피관리 시스템에서 사건보고가 LOCK되기전에 오버플로우될 수 있다. 그러므로 관리 시스템의 큐가 오버플로우되지 않도록 ThresholdTime을 설정하기 위해서는 다음식을 만족해야한다.

(정의 1)관리자의 큐가 오버플로우되지 않는조건.

$$\frac{E \left[ \sum_{i=1}^n C(\text{TimeQ}[i]) \right]}{\mu(Q_m) \cdot N} \leq t_i$$

이때, TimeQ 는 FULL.

$E \left[ \sum_{i=1}^n C(\text{TimeQ}[i]) \right]$  는 전체 TimeQ의 평균용량 (단위 : 패킷수).

$\mu(Q_m)$ 은 관리시스템의 평균서비스율 [단위 : 패킷수/sec] .

N은 노드수,  $0 < t_i < \infty$  이다.

이때,  $\lim_{\mu \rightarrow \infty} \frac{C}{\mu \cdot N}$  은 0이 되므로 oubThresholdTime이 무의미해져서 순수 사건보고기법 즉, 과보고 회피기

법을 이용하지 않는 형태가 되고,  $\lim_{\mu \rightarrow \infty} \frac{C}{N}$  는  $\infty$ 가 되어 발생하는 사건보고 마다 LOCK이 발생된다.

네트워크내에서 모든노드를 하나의 집합으로 보고 LOCK 발생시간이 일정하다고 가정하고 LOCK발생을 하나의 주기로 보게되면 LOCK사이클 시간  $t_{cycle}$  을 정의할 수 있다. 이때 LOCK이 발생하는 시스템 수  $N_{lock}$  을 정의할 수 있다.

[정의 2] 관리자큐가 임계치 H미만인 경우, 네트워크에서 LOCK 이 발생하는 시스템수 ( $N_{lock}$ )

$$\begin{aligned}
 N_{lock} &= \frac{RTT}{t_{cycle}} \\
 &= \frac{\tau_s + P_1 + P_2 + \tau_{dl}}{t_{cycle}} \\
 &= \frac{2(\tau_{dl} + P_d)}{t_{cycle}}
 \end{aligned}$$

RTT (Round Trip Time for UNLOCK)

- $\tau_s$  = 상태변경보고서비스의 전파지연시간
- $P_1$  = 상태변경보고서비스응답의 전송준비시간
- $P_2$  = M-SET(UNLOCK)의 전송준비시간
- $\tau_{dl}$  = M-SET(UNLOCK)의 전파지연시간
- (  $\tau_s = \tau_{dl}$  ,  $P_1 = P_2$ 로 가정)

이와같이  $RTT \geq t_{cycle}$ 과같은 조건이 발생하면 반드시 LOCK상태의 피관리 시스템이 발생하며, 이는 사

건보고 트래픽과 무관하게 된다. 그러므로  $t_{cycle}$ 의 값이 RTT보다 크도록 설정해야 한다. 관리시스템과 피관리 시스템사이의 PDU흐름은 다음 그림 4와 같다.

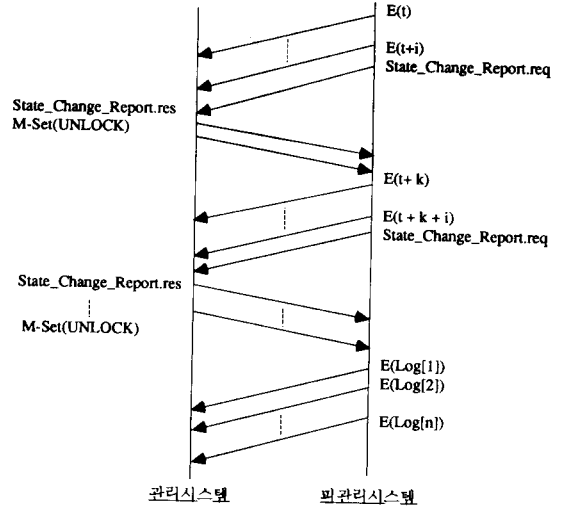


그림 4. PDU의 흐름도  
Fig. 4. Flow of PDU.

피관리 시스템의 LOCK 메커니즘을 가상코드로 나타내면 다음그림 5와 같다.

```

if (interrupted_signal == Event_Generated) {
    if (the_number_of_generated_event <= maxEventPerTime) {
        record the time of the current event ;
        send M-Event-Report ;
    }
    else
        calculate the difference of time between the latest
        event and the former event with maxEventPerTime value ;
    if (difference > ThresholdTime) {
        record the time of the current event ;
        send M-Event-Report ;
    }
    else
        send State_Change_Report.req ;
}
if (interrupted_signal == State_Change_Report.res) {
    if (Q_stat == GOOD)
        ThresholdTime = ThresholdTime - Adjustfactor ;
    else
        ThresholdTime = ResetTime ;
    if (interrupted_signal == Event_Generated)
        log the current event ;
    else if (interrupted_signal == UNLOCK)
        send the logged events ;
}

```

그림 5. 피관리시스템의 LOCK 알고리즘.  
Fig. 5. LOCK algorithm of a managed system.

Ⅳ. 시뮬레이션 모델

망관리 시스템에서 과보고를 회피하기 위한 기법은 하나의 관리시스템과 다수의 피관리 시스템사이의 상호작용에 의해 이루어진다. 1진 피드백 과보고 회피 기법은 다른 유사한 역할을 해주는 기법들과 비교가 되어야 하는데, 시뮬레이션 모델에서는 네트워크상의 Congestion Avoidance를 위해 주로 이용하는 이진 피드백 기법과 사건보고가 발생하는데로 바로 전송하는 순수기법이 비교된다.

○ 순수기법

피관리 시스템에서 사건보고가 발생하면 즉시 관리 시스템으로 해당사건 보고를 송신한다. 즉, 과보고를 회피하기 위한 어떠한 기법도 이용되지 않는다.

○ 이진 피드백 기법

과보고를 회피하기 위하여 관리시스템은 M-Set PDU를 이용하여 LOCK과 UNLOCK 2가지 상태를 지정할 수 있다. 이때 관리시스템의 큐상태에 따라 LOCK과 UNLOCK을 지정하며 피관리시스템은 로깅을 위한 관리객체를 유지하지 않는다.

○ 1진 피드백 과보고회피 기법

과보고를 회피하기 위하여 피관리시스템은 ThresholdTime에따라 자신을 LOCK상태로 유지시키며 관리시스템은 이를 해제하는 M-Set PDU를 전달하여 UNLOCK 시킨다. 기본적으로UNLOCK

만을 이용하며 암시적 LOCK을 갖게된다. LOCK 상태에서 발생한 사건보고는 로그라는 관리객체에 유지된다.

시뮬레이션에서 사용되는 도구는 MIL3사의 OPNET이라는 네트워크 전용 시뮬레이터를 이용한다. OPNET은 그 구성상 3가지 모델을 정의하여야 하는데, 이들은 네트워크 모델, 노드모델, 프로세스 모델등이다. 이와같이 과보고 회피기법에 대한 시뮬레이션 모델은 3단계로 나누어 설명될 수 있다.

1. 네트워크 모델

순수기법, 1진피드백 기법, 이진 피드백 기법을 위한 네트워크 모델로 그림 6과 같이 Ethernet을 선택 하였다.

Ethernet에 연결된 노드는 과보고를 통제하는 관리 시스템과 사건보고를 발생시키는 피관리 시스템으로 구성된다. 10개의 피관리 시스템이 1개의 관리 시스템으로 사건보고를 행하는 네트워크 모델이 설정된다.

2. 노드모델

네트워크 모델에서 Ethernet에 연결된 노드는 관리 시스템과 피관리시스템의 2가지 형태가 있다. 이것은 노드모델에서 정의 되는데 우선 피관리 시스템에 대한 노드 수준의 모델은 그림 7과 같다.

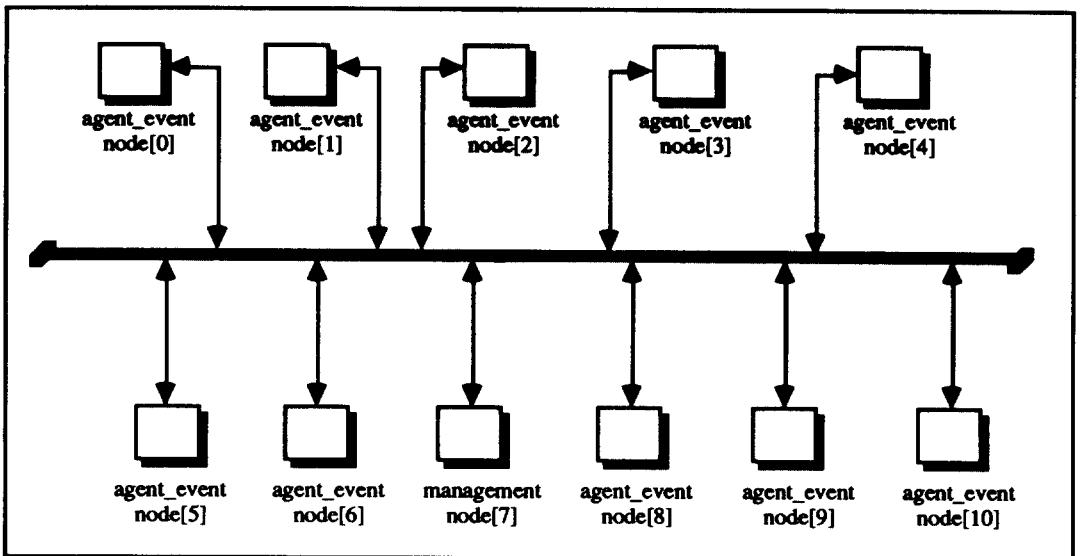


그림 6. 네트워크 모델  
Fig. 6. Network model.

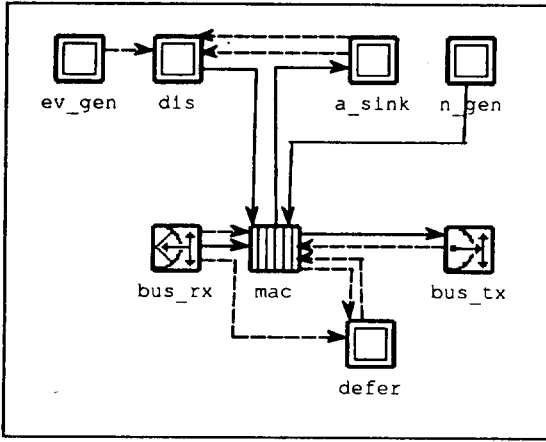


그림 7. 피관리 시스템의 노드모델  
Fig. 7. Node model of a managed system.

그림에서와 같이 사건보고에 대한 피관리 시스템은 ev\_gen, dis, a\_sink, n\_gen, bus\_rx, mac, defer, bus\_tx 모듈들로 구성된다. bus\_rx, mac, bus\_tx, defer 모듈은 OPNET에서 Ethernet을 구현하기 위해 지원되는 모듈이고 나머지 모듈들은 상위계층 즉, 과보고 회피를 위해 제안된 모듈들이다. 실선은 PDU의 흐름을 나타내며 점선은 신호의 흐름을 의미한다. ev\_gen은 사건보고를 발생시키는 모듈로서 시뮬레이션의 시작을 알리는 인터럽트가 발생하면 관리 시스템으로의 사건보고를 시작한다. 이때, 순수 기법은 어떠한 통제기법이 이루어 지지않으므로 바로 MAC 계층에 접근하며, 이진 피드백 기법은 관리시스템으로부터 LOCK신호가 수신될 때까지 순수 기법과 동일하게 동작한다. 1진 피드백 기법의 경우 모든 사건보고는 사건보고 판별자인 dis모듈을 경유하게 되는데 이부분에서는 그림 5에서 설명된 LOCK 알고리즘이 적용된다. n\_gen 모듈은 일반 패킷의 발생을 위해 존재하는 모듈이다. 이와같이, 피관리 시스템은 사건보고를 위한 관리 패킷과 응용계층의 일반패킷을 발생시킨다. mac으로부터 유입된 패킷은 a\_sink 모듈로 전달하게 된다. a\_sink모듈은 관리시스템으로부터 유입되는 패킷들로부터 관리 시스템의 정보를 입수하여 사건보고 판별자에게 제공한다. a\_sink 모듈에서 제공하는 기능은 다음과 같다.

- 0 일반패킷의 소거
- 0 UNLOCK 패킷에 대한 정보제공
- 0 M-EVENT-REPORT.confirm 패킷에 대한 처리

이와달리 관리 시스템은 그림 8과 같이 모델링 된다.

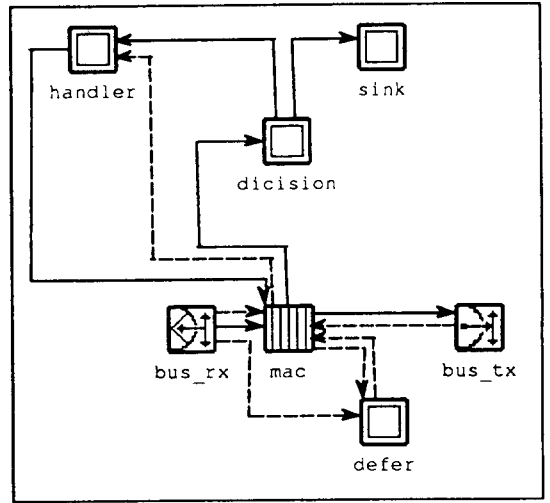


그림 8. 관리 시스템의 노드모델  
Fig. 8. Node model of a management system.

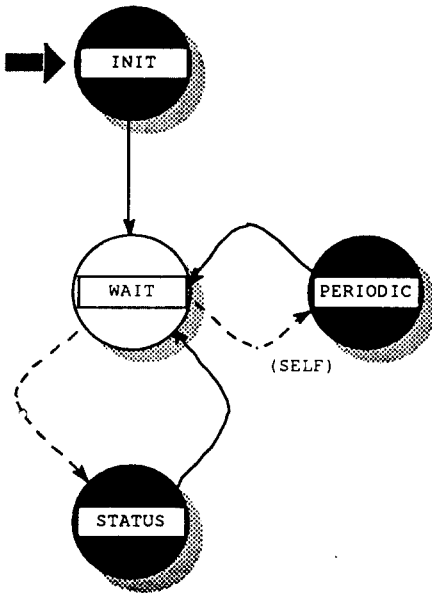
bus rx, mac, defer, bus\_tx는 피관리 시스템의 경우와 동일하게 OPNET에서 제공하는 모듈을 이용하였으며 handler, sink, decision 모듈이 과보고 회피를 위해 구성되었다. 우선 1진 피드백 기법을 살펴보자. handler모듈은 LOCK 신호를 나타내는 M-EVENT-REPORT.request PDU 를 수신하면 M-EVENT-REPORT.confirm PDU를 송신 피관리 시스템으로 전송한다. 그후 그림 4에서 설명된 UNLOCK 알고리즘에 따라 자신의 큐상태를 점검하고 상태가 좋으면 바로 UNLOCK 패킷에 해당하는 M-SET PDU를 송신하고 상태가 좋지 않으면 송신 피관리 시스템의 ID를 EventReportDisabledQ에 기록한다. 이때, 관리 시스템의 큐는 M/M/1으로 모델링 된다. decision모듈은 수신 사건보고가 일반 사건보고인지 LOCK을 나타내는 사건보고인지를 판단하여 LOCK처리를 위한 handler모듈이나 패킷을 파기시키는 sink 모듈로 경로 설정한다. 순수기법의 경우 관리 시스템은 아무런 제어동작을 하지 않는다. 이진 피드백 기법의 경우 관리 시스템의 큐상태에 따라 LOCK이나 UNLOCK에 해당하는 M-SET PDU 를 Ethernet의 모든 피관리 시스템에게 전송하게 된다. 이와같이, 관리시스템과 피관리 시스템 모델은 동작하기 위한 가정을 갖게되는데 노드 모델에서 주어진 가정은 다음과 같다.

- PDU의 크기 (지수분포) m : 2048bits
- M/M/1의 평균서비스율  $\mu$  : 1Mbps
- 전파지연시간  $\tau$  :  $2 \times 10^6$ m/s



3. 프로세스 모델

노드모델에서 설정된 모듈들은 그내부 처리구조를 명시하기 위해 프로세스 모델이 필요하며 노드모델의 각 모듈은 대응하는 프로세스 모델을 갖고있다. 프로세스모델은 FSM(Finite State Machine)이라는 정규형식으로 표현되며 n\_gen, ev\_gen, a\_sink, sink, decision등은 간단한 구조를 갖고있다. 과보고 회피기법의 핵심적인 역할을 담당하는 handler 와 dis프로세스 모듈은 다음과 같다. 우선 handler 프로세스 모듈에 대한 FSM은 다음 그림 9 (a)와 같으며 이모듈의 헤더 블록 속성은 그림 9 (b)와 같다. 이때 실선은 무조건 분기를 의미하며 점선은 조건분기를 의미한다.



(a)

```
#define SELF op_intrpt_type() == OPC_INTRPT_SELF
#define ALERT_DISABLE op_intrpt_type() == OPC_INTRPT_STRM

#define CHECK_INTERVAL 0.1
#define NODE_NUM 10
#define STAT_INSTREAM 0
#define IN_STREAM 0
#define OUT_STREAM 0
#define MAC_BUFFER_BIT_SIZE 5000000
#define HIGH_WATER_MARK 0.7
#define LOW_WATER_MARK 0.3

int load_num = 0;
double wide;
```

(b)

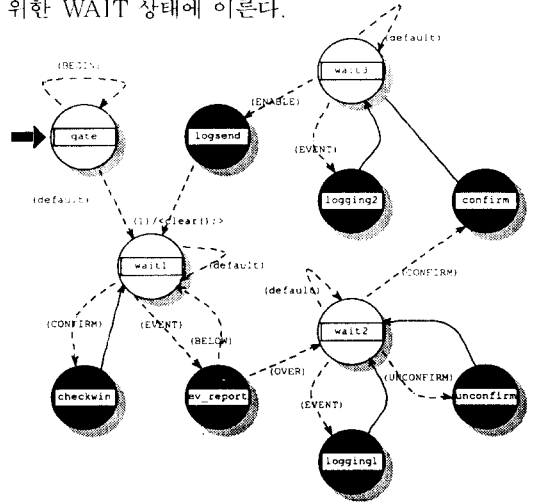
그림 9. (a) handler 프로세스 모델

(b) handler 프로세스 모델의 헤더블럭 속성

Fig. 9. (a) Handler process model.

(b) Header block attribute of handler process model.

초기화를 거친후 원격지의 피관리 시스템으로부터 LOCK을 알리는 M-EVENT-REPORT.request가 유입되든가 혹은 관리시스템 큐의 주기적인 검사를 위한 WAIT 상태에 이른다.



(a)

```
#define MaxEventPerTime 40
#define LOGSIZE 80
#define DOWNSTREAM 0
#define EVENTGENERATED 0
#define ENABLED 1
#define FEEDBACK 2
#define CONFIRMTIME 4.0
#define MANAGERNUM 7
#define GOOD 1.0
#define MINIMUM 0.12
#define BEGIN op_intrpt_type() == OPC_INTRPT_BEGSIM
#define EVENT op_intrpt_type() == OPC_INTRPT_STAT &&\
    op_intrpt_stat() == EVENTGENERATED
#define ENABLE op_intrpt_type() == OPC_INTRPT_STAT &&\
    op_intrpt_stat() == ENABLED
#define CONFIRM op_intrpt_type() == OPC_INTRPT_STAT &&\
    op_intrpt_stat() == FEEDBACK
#define UNCONFIRM op_intrpt_type() == OPC_INTRPT_SELF
#define BELOW flag == 0
#define OVER flag == 1
```

(b)

그림 10 (a) dis 프로세스 모델

(b) dis 프로세스 모델의 헤더블럭 속성

Fig. 10. (a) dis process model.

(b) Header block attribute of the

dis process model.

LOCK을 알리는 신호가 유입되면 M-EVENT-REPORT.confirm PDU를 송신 피관리 시스템에게 전송하고 UNLOCK 알고리즘에 따라 관리 시스템의 큐를 검사하기 위해 STATUS로 천이한다. 이때, HIGH WATER MARK와 LOW WATER MARK라는 임계치(threshold value)를 참조하는데 이는 관리시스템 큐의 히스테리시스 특성을 반영하기 위해서 존재한다. 상태 검사후 피관리 시스템에

게 UNLOCK을 알리는 M-SET PDU 를 전송하고 WAIT상태로 천이한다. 큐상태가 좋지 않은 경우 주기적으로 큐를 검사하기 위해 인터럽트가 걸리게 되는데 이때 PERIODIC 상태로 천이하게 된다. 이상태에서는 LOCK상태의 피관리 시스템에게 UNLOCK을 전송하여 사건보고가 재개할 수 있게 한다. 피관리 시스템에게 중요한 역할을 담당하는 dis 프로세스 모듈은 그림 10(a)와 같으며 이 모듈의 헤더블럭 속성은 그림 10(b)와 같다.

초기화를 거친후 wait1과 ev\_report 상태들은 국부적으로 발생하는 사건보고를 처리한다. Max EventPerTime의 검사결과가 BELOW인지 OVER인지를 결정하여 wait2상태로 천이한다. 이상태에서 발생하는 사건보고는 로그에 저장된다. 관리 시스템으로부터 CONFIRM을 받게되면 confirm 상태로 천이하며 ThresholdTime 값을 변경할 준비를 하고 wait3상태로 천이한다. 이상태에서 발생하는 사건보고도 로그에 역시 저장된다. 이어서 UNLOCK을 알리는 M-SET PDU를 수신하면 국부적으로 로그에 저장된 사건보고를 관리 시스템으로 송신한다.

V. 결과 및 고찰

피관리 시스템의 수를 10개의 노드로 설정한 Ethernet에서 1진 피드백과보고회피기법의 특성을 살펴보고, 이를 이진 피드백 기법, 순수기법등과 비교하였다. 시뮬레이션 시간이 지남에따라 나타나는 ThresholdTime의 변화는 1진 피드백 기법이 사건 보고 부하에 적응하고 있음을 나타내고 있는데 그림 11은 이를 보여준다. 관리 시스템의 상태가 좋은 경우 ThresholdTime은 계속 감소하고 최저치까지 감소한후 그값을 유지하다가 상태가 나빠지면 다시 재설정되어 이전의 상태를 반복하게 된다. 이와같이 이 그림은 1진피드백 기법이 적절한 운영점으로 수렴하고 있음을 보여준다. 이 기법에서 중요한 고려사항중의 하나가 공정성을 보장하는가인데, 이는 네트워크 계층에서의 혼잡 제어에서도 중요한 평가기준의 하나가 되며 피관리 시스템 모두에 대해 나타나는 공정성은 그림 12와 같다. 이 그림이 보여주듯이 모든 노드가 거의 동일한 값으로 공정성을 유지하고 있음을 알 수 있다. 그림 13과 14는 사건보고 부하에따라 나타나는 평균활성화 노드수와 임의의 노드에서 발생하는 LOCK 유지시간을 보여주고 있으며, 이는 1진피드백 기법의 민감도를 알아보기 위한 것이다. LOCK 시간에 있어서는 사건보고부하가 0.8정도에서 증가하기 시작한다. 이때 TimeQ와 로그큐의 크기가 증가하면

더 빠른 속도로 LOCK시간이 증가함을 알 수 있다.

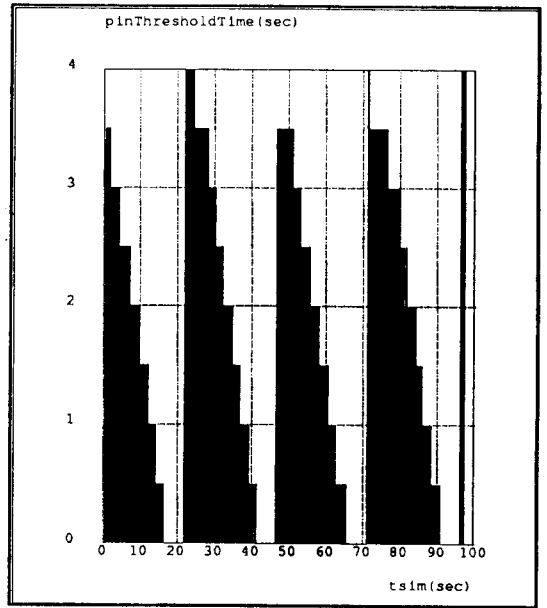


그림 11. 시뮬레이션 시간에 따른 ThresholdTime의 변화

Fig. 11. Changes of ThresholdTime value according to the simulation time.

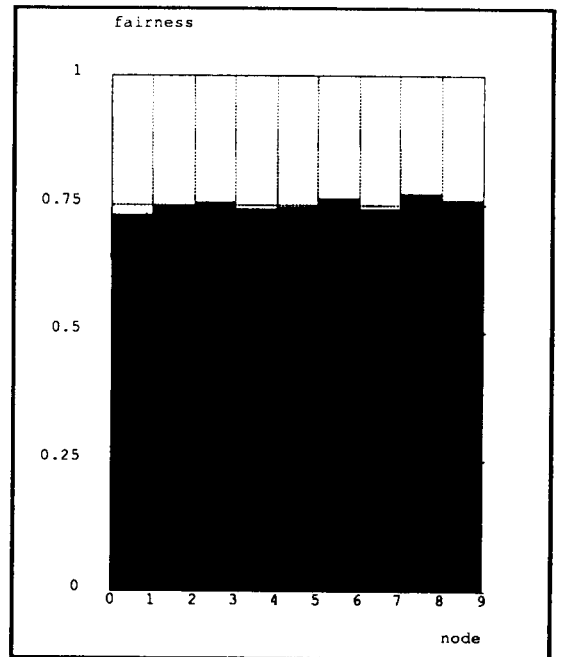


그림 12. 각 노드에 대한 공정성

Fig. 12. Fairness for each node.

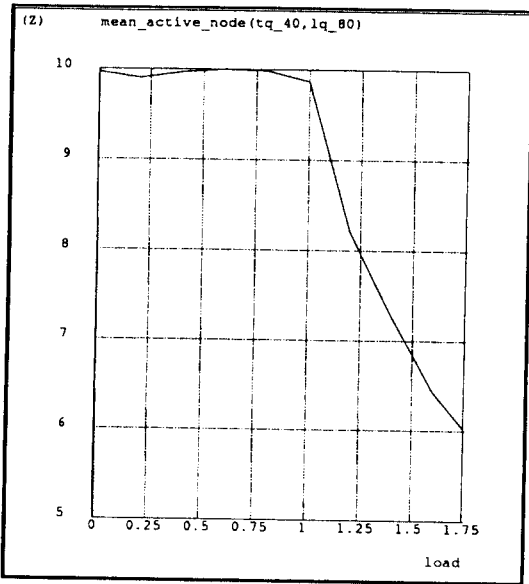


그림 13. 부하에 따른 평균 활성화 노드수  
Fig. 13. Number of mean active nodes according to the load.

즉, 일정부하 이상인 경우, 로그큐와 TimeQ의 증가는 오히려 LOCK이 걸려있는 시간을 증가시키며, 이는 UNLOCK시 로그큐에 있던 사건보고 전송에 기인된 것으로 분석된다. 그리고, 평균 활성화 노드수에 있어서 살펴보면 사건보고부하가 아주 큰 경우 그 수가 급격하게 감소하며 사건보고부하가 1정도까지는 모든 노드가 균등하게 활성화 되어있음을 알 수 있다. 이는 부하가 큰 경우 활성화 노드수를 제어하여 관리자 시스템의 부하를 최소화 하고 있음을 나타낸다. 그림 15, 16, 17은 일반 패킷의 부하가 증가함에 따라 나타나는 패킷 분실률과 관리자 큐의 지연시간, Goodput을 나타낸다. 또, 이를 사건보고부하가 0.6, 0.8, 1.0으로 증가 되었을때의 상태를 나타내었다. 이때, Goodput은 요구한 사건보고전송량  $D_i$ 에 대한 성공적으로 수신한 사건보고량  $A_i$ 가 된다. 발생한 사건보고는 국부적인 원인 즉, 큐의 오버플로우, 용량초과 등에 의하여 영향을 받게된다.

$$Goodput = \sum \frac{A_i}{D_i} \quad (i \text{는 임의의 피관리 시스템})$$

우선 패킷 분실률의 경우 사건보고부하가 0.6인 경우 일반패킷부하가 0.4일때까지는 패킷손실이 전혀 일어나지 않으며 일반패킷의 부하가 증가할수록 사건보고에 대한 손실은 급격히 증가하게 된다.

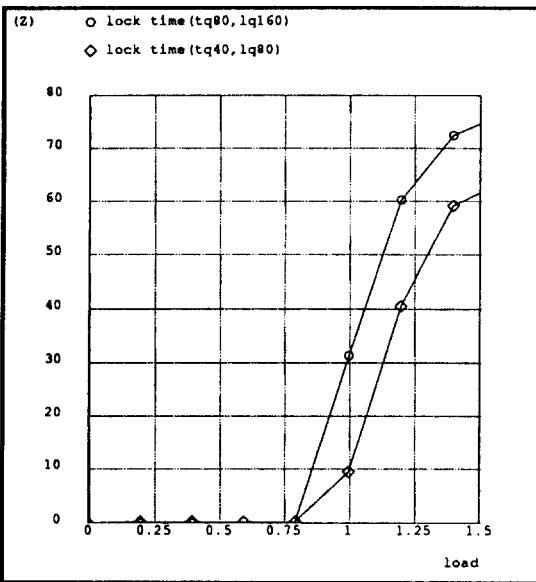


그림 14. 부하에 따른 LOCK 시간  
Fig. 14. LOCK times according to the load.

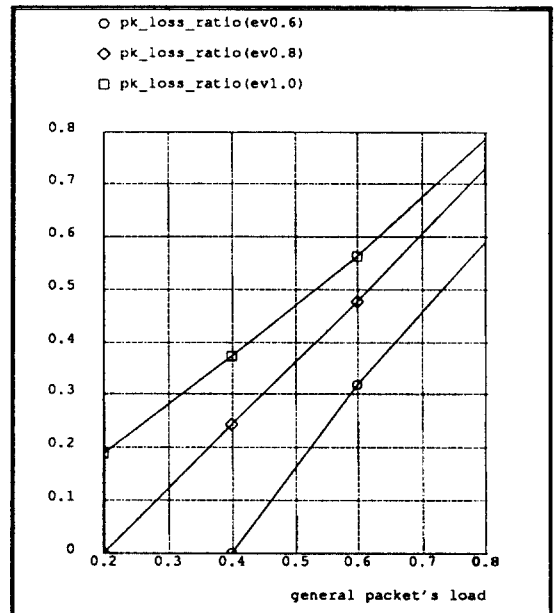


그림 15. 일반 패킷 부하에 따른 패킷 분실율  
Fig. 15. packet loss ratio according to the load of general packet.

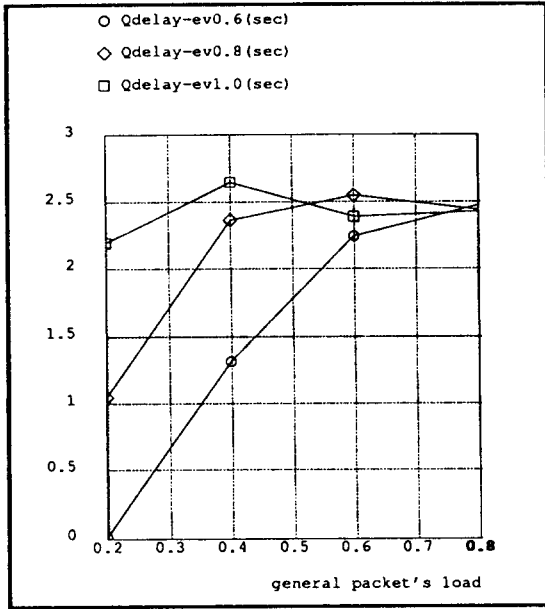


그림 16. 일반 패킷 부하에 따른 관리자 큐 지연시간

Fig. 16. Delay of a manager queue according to the load of general packet.

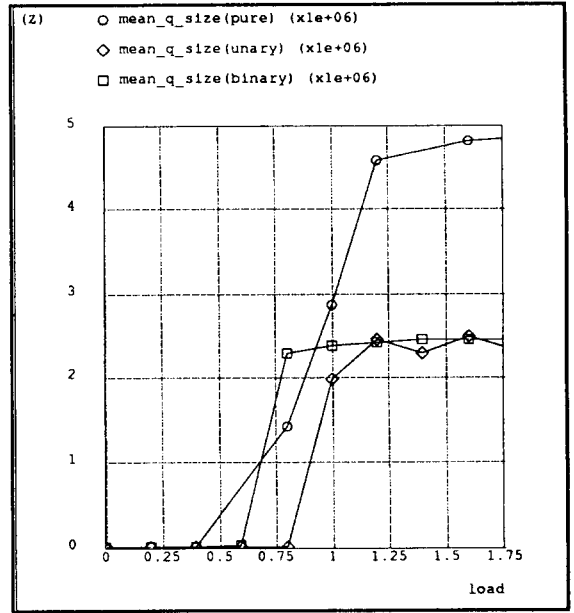


그림 18. 각 기법에 있어서 부하에 따른 평균 큐 용량 비교

Fig. 18. Comparison of mean queue size according to the load in each scheme.

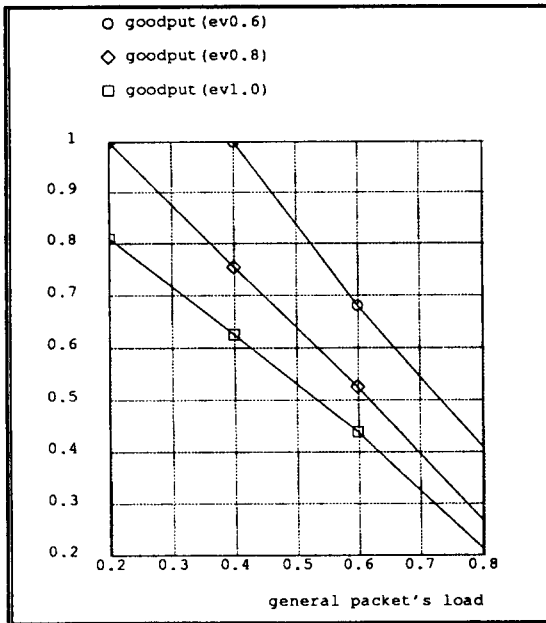


그림 17. 일반 패킷 부하에 따른 goodput

Fig. 17. Goodput according to the load of general packet.

이는 사건보고부하가 0.8, 1.0으로 증가해도 동일한 현상을 보인다. 다시말해서 1진피드백 기법은 사건보고율과 일반패킷의 부하에 민감하게 반응함을 알 수 있다. 관리 시스템의 큐 지연시간 면에서는 사건보고 부하가 0.6, 0.8, 1.0으로 증가하고 일반패킷부하가 증가해도 2.5sec 정도에서 더이상 증가하지 않는다. 이는 관리시스템의 부하를 1진피드백 기법이 통제하고 있음을 보여준다. 즉, 패킷손실이 피관리 시스템에서 발생하여 관리시스템에서는 과부하 현상이 발생하지 않는다. Goodput은 패킷 손실 확률과 밀접한 관계를 갖는데 그림 17에서와 같이 사건보고율과 일반패킷부하의 증가는 Goodput에 많은 영향을 주고 있음을 알 수있다. 한편 1진 피드백 기법, 이진 피드백 기법, 순수 기법을 평균 큐용량, 평균 큐지연시간, Goodput에 대하여 상호 비교 분석 하였다. 그림 18에서는 사건보고부하에 따라 나타나는 평균 큐용량을 보여주고 있는데 순수기법은 사건보고부하에 민감하게 반응하며 많은양의 큐용량을 필요로하고 있으며 1진 피드백 기법과 이진 피드백 기법은 상대적으로 적은양의 큐용량을 필요로 한다. 그러나, 1진 피드백 기법이 사건보고부하에 적응력이 좋아서 덜 민감함을

알 수 있다. 큐 지연시간면에서도 이 기법이 가장 우수함을 알 수 있는데 이는 피관리 시스템에서 사건보고를 통제함으로써 인해 발생한다. 그림 19는 이를 나타낸다. 그림에서도 나타나듯이 1진피드백 기법은 부하에 있어서 다른 기법에 비해 상대적으로 적응력이 우수하며 지연 시간 측면에서도 다른 기법에 비해 낮은 값을 갖는다. 그러나, Goodput면에서는 순수기법이 가장우수하게 나타나며 1진 피드백 기법이 어느 정도 우수하고 이진 피드백 기법이 가장좋지를 못하다. 그림 20에서 알 수 있는것은 1진피드백 기법이 과부하 상태에서 순수기법보다 성능이 저하됨을 알 수 있다. 순수기법은 관리 시스템에서 오버플로우가 발생하여 계속적으로 발생하는 사건보고가 모두 손실되며 통제가 가능한 1진 피드백 기법은 피관리시스템에서만 패킷의 손실이 발생한다. 이러한 차이에도 불구하고 1진피드백 기법의 Goodput이 감소하는 이유는 LOCK메카니즘에따른 RTT [Round Trip Time] 에 기인한다. 또, 순수기법은 과부하상태에서 관리시스템에게 피관리 시스템의 과부하를 전가시키며 이에비해 1진 피드백 기법은 피관리 시스템의 과부하를 관리시스템으로 전달하지 않는다.

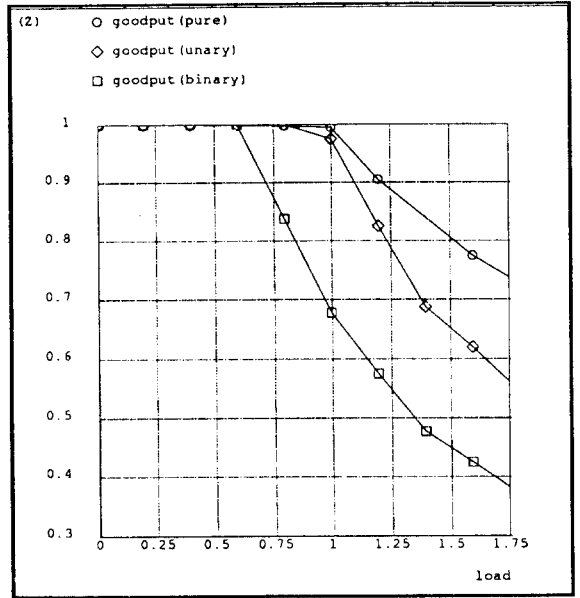


그림 20. 각 기법에 있어서 부하에 따른 Goodput 비교

Fig. 20. Comparison of goodput according to the load in each scheme.

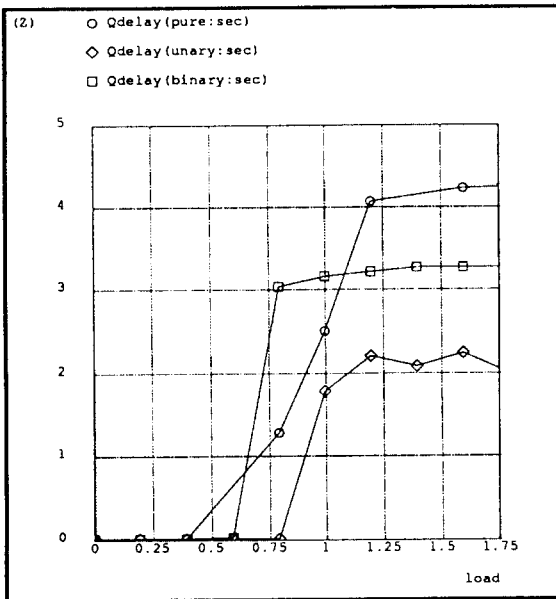


그림 19. 각 기법에 있어서 부하에 따른 큐 지연 시간 비교

Fig. 19. Comparison of mean queueing delay according to the load in each scheme.

## VI. 결론

본 논문에서는 OSI 망관리 시스템에서 피관리 시스템이 관리 시스템으로 사건보고를 하는 경우 트래픽 집중현상이 발생함으로써 인한 관리 시스템의 과부하를 회피하기위한 기법인 1진피드백 과보고 회피기법을 제안하였다. 제안한 1진 피드백 과보고 회피 기법의 시뮬레이션 모델을 설정하고 이를 통해 여러가지 특성을 검출 하였으며, 망관리에 적용할수있는 유사기법인 이진 피드백 기법, OSI 망관리시스템의 순수기법과 상호 비교 분석하였다.

이 기법은 피관리 시스템 자신이 LOCK되었음을 통보할때 관리시스템은 자신의 상태가 좋은경우 이를 해제하기 위한 UNLOCK을 해당 피관리 시스템에게 전송하는 원리로서, 관리시스템은 자신에게 주어질 임계값을 이용하여 통제하는 방식으로, 그리고 피관리시스템은 관리 시스템의 부하에 적응하는 감소/재설정 알고리즘을 이용하여 전체적인 사건보고의 트래픽을 효율적으로 관리한다. 이 기법의 특성을 알아보기 위해 ThresholdTime, 공평성, LOCK유지 시간, 평균 활성화 노드수 등을 측정하였다. 그리고 일반 패킷과의 관계를 알아보기위해 패킷 분실확률, 관

리 시스템의 큐 지연시간, Goodput등을 검출하였다.

그 결과 공평성의 보장과 더불어 특히 관리시스템의 과부하를 완벽하게 통제 할 수있음을 확인 하였다. 그리고 사건보고와 일반 트래픽 부하의 합이 1.0 이하인 경우 파라미터 특성이 모두 우수하게 나타났으며 부하의 합이 매우 높은경우 이들에 대한 성능이 저하됨을 알수있었다. 또한 제안 기법과 다른 2가지 유사기법의 평균큐용량, 큐 지연시간, Goodput특성을 상호 비교 분석하여 본 논문에서 제안한 1진 피드백 과보고 회피기법의 우수성을 입증하였다.

추후 1진피드백 과보고 회피기법의 성능향상에대한 연구가 계속 되어야 할 것이다

#### 參 考 文 獻

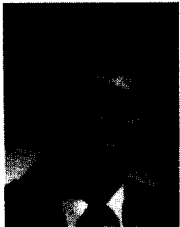
- [1] Information Processign systems - Open Systems Interconnection, "System Management Overview," ISO/IEC 10040, 1990.
- [2] Information Processing Systems - Open Systems Interconnection, "Common Management Information Service Definition," ISO/IEC 9595, 1991.
- [3] Information Processing Systems - Open Systems Interconnection, "Common Management Information Protocol Definition," ISO/IEC 9596, 1991
- [4] Information Processing Systems - Open Systems Interconnection, " Systems Management, Part 1 : Object Management Function", ISO/IEC 10164-1, 1990.
- [5] Information Processing Systems - Open Systems Interconnection, " Systems Management, Part 2 : State Management Function", ISO/IEC 10164-2, 1990.
- [6] Information Processing Systems - Open Systems Interconnection, " Systems Management, Part 5 : Event Report Management Function", ISO/IEC 10164-5, 1990.
- [7] Information Processing Systems - Open Systems Interconnection, " Systems Management, Part 6 : Log Control Function", ISO/IEC 10164-6, 1990.
- [8] L. Steinberg, "Techniques for Managing Asynchronously Generated Alerts", RFC1224, May 1991.
- [9] K. K. Ramakrishnan, R. Jain, "A Binary Feedback scheme for Congestion Avoidance in Computer Networks", ACM Transactions on Computer Systems, vol. 8, no. 2 pp. 158-181, May 1990.
- [10] K. K. Ramakrishnan, R. Jain, "A Binary Feedback scheme for Congestion Avoidance in Computer Networks with a connectionless Network Layer", ACM SIGCOMM'88 pp. 303-313, 1988.
- [11] D. M. Chiu, R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in ccomputer Networks", Computer Networks and ISDN Systems, no. 17, pp. 1-14 1989.
- [12] Van Jacobson, "Congestion Avoidane and Control", ACM SIGCOMM'88 pp. 314-329, 1988.
- [13] Adrian E. Eckberg, Jr, "The single Server Queue with Periodic Arrival Process and Deterministic Service Times", *IEEE Transactions on Communications*, vol. 27, no. 3 pp. 556-562, march 1979.
- [14] Y. T. Wang, B. Sergupta, "Performance Analysis of a Feedback Congestion Control Policy", ACM SIGCOMM'91 pp. 149-157, 1991.
- [15] S. Shenker, "A Theoretical Analysis of Feedback Flow Control", *Computer Communication Review*, vo. 20 pp. 156-165, 1990.
- [16] J.K.Ryu, H.J. Kang, J.W. Chung, S. J. Ahn, O.H. Byeon, "The Design and Implementation of Systems Management Application Service Element [SMASE] for OSI Network Management", JWCC-7 pp. 349-356, July 1992.
- [17] J. H. Ahn, H. J. Kang, J.W. Chung, S.J. Ahn, O.H. Byeon, "The Design

and Implementation of MIT for Management Information Base". JWCC-7 pp. 357-564, July 1992.

[18] 변옥환, 진용옥, "OSI 망 관리 프로토콜의 트래픽 제어에 대한 성능분석", 한국통신학회논문지, 제 17권 7호 P758-766, 1992년 7월.

[19] 변옥환, 진용옥, "표준망관리 프로토콜의 Scope/Filter에 대한 성능분석", 전자공학회 논문지, 제 29권 A편 7호 P10-18, 1992년 7월.

著 者 紹 介



邊 玉 煥(正會員)

1953年 8月 28日生. 1979年 2月 한국항공대학 통신 공학과(공학사) 1985年 인하대학교 대학원 전자공학과(공학석사). 1983年 3月 ~ 현재 경희대학교 대학원 전자공학과(박사과정수료).

1978年 9月 ~ 현재 KIST시스템공학연구소 연구전산망 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.