

## 분산형 동기식 전송 관리망에서의 시험관리를 위한 concurrency control 알고리즘 제안 및 성능분석

正會員 李玟炯\*\*, 朴明桓\*, 宋重求\*, 曹圭燮\*, 朴炳哲\*

### Algorithm and Performance Analysis of Concurrency Control for the Test Management in the Distributed SDH Management Network

Min Hyoung Lee\*\*, Myeong Hwan Park\*, Joong Goo Song\*  
Kyu Seob Cho\*, Byung Chul Park\* Regular Members

#### 要 約

개방형 분산 환경을 바탕으로 하는 분산 관리망의 핵심은 독립적으로 관리행위를 수행하는 다수의 관리자들이 존재한다는 점이다. 이러한 관리자들은 관리행위를 수행할 때 일관성을 유지하기 위하여 관리자원에 lock을 걸어주며 이 과정에서 발생할 수 있는 deadlock은 분산 관리망의 성능저하 요인으로 작용한다. 따라서 deadlock이 발생하는 것을 방지하기 위한 방안으로 concurrency control을 위한 특별한 알고리즘들이 연구되고 있다.

본 연구에서는 분산 관리망에서 관리자들간에 관리정보를 교환함으로써 불필요한 관리정보 전송 절차를 줄이고 deadlock을 해소하는데 소요되는 시간지연을 최소화할 수 있는 새로운 concurrency control 알고리즘인 관리자간 협상 알고리즘을 제안하였다. 그리고, 제안한 알고리즘의 성능을 평가하기 위하여, 기존 알고리즘과 제안한 알고리즘을 분산형 동기식 전송 관리망에서의 시험관리에 각각 적용하고 이에 대한 시뮬레이션을 수행하였다.

#### ABSTRACT

The key concept of distributed management network is that there exists many independent managers each of which has its own management capacity. Management resources are locked by a manager in order to maintain the consistency in management procedure. But sometimes the locking mechanism gives rise to the deadlock which cause effects on the performance degradation

\* 성균관대학교 전자공학과  
Dept. of Electronic Engineering, Sung Kyun Kwan Univ.  
\*\*대우통신연구소  
DaeWoo Telecom Research Institute  
論文番號 : 95098-0310  
接受日字 : 1995年 3月 10日

of the distributed management network. The deadlock can be prevented using the algorithms implementing the concurrency control. The conventional algorithms for concurrency control can be adapted to the distributed management network but they does not fully use the transmission resources given by the DCN(Data Communication Network).

This paper presents a new concurrency control algorithm, the InterManager Negotiation Algorithm(IMNA), which takes advantage of the DCN thoroughly including the OS to OS communication capacity. The new algorithm minimizes the additional messages with concurrency control, thus, reduces processing delays significantly while sustaining the advantages of conventional algorithms. Two types of the test management models were applied for the SDH management network in distributed management environment; one based on the conventional concurrency control algorithm, wound-wait algorithm, and the other based on the IMNA proposed in the paper. Simulations were achieved for the evaluation of applied test management models.

## I. 서 론

기존의 비동기식 전송망이 동기식 전송망으로 진화하면서 ITU-T(International Telecommunication Union-Telecommunication Standardization sector)는 이를 관리하기 위한 방안으로 통신 관리망(TMN: Telecommunications Management Network)에 근거한 동기식 전송 관리망(SMN: SDH Management Network)을 국제표준으로 권고하고 있다[1]. TMN은 통신망의 종류와 이에 관련된 설비 또는 서비스 등에 관계없이 망관리를 수행할 수 있는 환경을 제공하며 여기서 수행하는 모든 관리행위는 OSI(Open Systems Interconnection) 표준에 따라 이루어진다[2].

관리망의 형태는 통신망의 규모와 특성, 지리적 여건 등을 감안하여 구성되어지며 최근의 망관리 시스템은 컴퓨터 관련 기술의 발달과 개방형 분산 환경(open distributed environment)의 도입에 따라 초기의 단일 관리 시스템에 의한 집중형 관리망의 형태에서 여러개의 관리 시스템으로 구성된 분산형 관리망의 형태로 발전되는 추세이다[3][4].

분산 관리망에서는 다수의 관리자(manager)들이 주어진 관리기능을 독립적으로 또는 공동으로 수행한다. 이때 관리자들은 관리행위의 일관성(consistency)을 유지하기 위해서 관리대상에 lock을 걸어주는데 이 과정에서 발생할 수 있는 deadlock은 분산 관리망의 성능저하 요인으로 작용한다. 이러한 deadlock이 발생하는 것을 방지시켜 주는 방안으로 concurrency con-

trol을 위한 특별한 알고리즘들이 많이 연구되고 있다 [5].

기존에 제안된 concurrency control 알고리즘들은 관리자와 관리대상간에서 수행되는 것을 기본으로 하나, 본 연구에서는 분산 관리망에서 관리자들간에 concurrency control을 위한 메시지물 교환함으로써 관리대상에 부담되는 부하 및 전송 절차를 줄이고 deadlock을 해소하는데 소요되는 시간지연을 최소화할 수 있는 새로운 concurrency control 알고리즘인 관리자간 협상 알고리즘을 제안한다.

제안한 알고리즘의 성능평가를 위하여, 분산형 동기식 전송관리망에서 이루어지는 시험관리에, 제안한 알고리즘과 기존 알고리즘 중에서 우수하다고 평가된 wound-wait 알고리즘을 적용하고 이에 대한 시뮬레이션은 OPNET을 사용하여 수행하였다.

## II. 기존 Concurrency Control의 문제점

OSI의 망관리 측면에서 보았을 때 관리대상(MO:Managed Object)들은 기능적, 기술적, 지역적 기준에 따라 일정한 관리영역(management domain)에 포함되며, 이러한 관리영역들은 독립적으로 수행되는 각각의 관리자들에 의해 관리되어 결국 그림 1과 같은 분산형 관리망의 개념을 형성한다.

일반적으로 관리대상들은 그림 1과 같이 기능적 또는 지역적으로 관리 영역이 중첩된 부분에 존재하거나 하나의 관리행위가 여러개의 관리영역에 걸쳐 수행되어야 하는 경우가 발생할 수 있으며, 이 경우에 두개 이상의 관

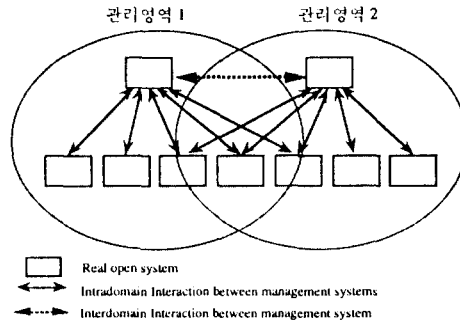


그림 1. 분산형 관리망의 구성도

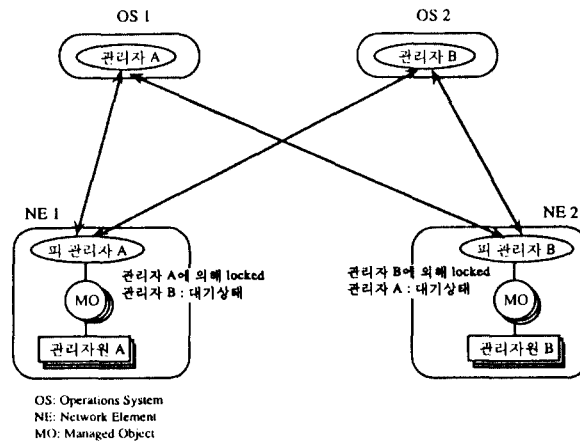


그림 2. Deadlock이 발생하는 예

리자들이 관리대상들을 공유하는 형태가 된다. 따라서 둘 이상의 관리자들이 동시에 같은 관리자원들을 사용하면서 특정 관리행위들을 수행하려고 할 경우에는 관리자원 사용권리를 어느 관리자가 가질 것인가가 문제된다.

이와 같은 문제점을 해결하기 위하여 resource-locking 메커니즘이 도입되었다. 이 메커니즘을 사용하여 관리자는 관리행위 수행동안 필요한 관리자원 사용권을 유지할 목적으로 관리자원에 lock을 걸어준다. 이 경우 다른 관리자는 lock이 걸린 관리자원을 액세스할 수 없게 되고 관리 환경의 일관성이 유지될 수 있다. 그

리나, 하나의 관리행위가 수행되기 위해서 둘 이상의 관리자원들이 사용될 경우에는 resource-locking 메커니즘으로 인하여 deadlock(deadly embrace)이 발생할 수 있는데 그 예를 그림 2에 나타내었다.

그림 2는 두 개의 관리자 A, B가 관리행위 수행을 위해 동시에 두 개의 똑같은 관리자원을 요구할 경우 발생할 수 있는 deadlock 상태를 보여주고 있다. 관리자 A는 피관리자 A의 관리자원 A에 대해서는 lock을 걸어서 사용권리를 얻었고 관리자 B에 의해 lock이 걸린 피관리자 B의 관리자원 B에 대한 사용권리를 요구하고

있다. 한편 관리자 B는 피관리자 B의 관리자원 B에 대한 사용권리를 얻고 관리자 A에 의해 lock이 걸린 피관리자 A의 관리자원 A에 대한 사용권리를 요구하고 있다. 결국 관리자들은 관리행위 수행에 필요한 두 관리자원들을 모두 확보하지 못하게 되며, 서로에게 걸린 논리적인 lock을 풀어주지 않는 한 영원히 locking 상태에 머물러 있게 된다. 이러한 경우를 deadlock이라고 하며 이 상태를 해결하기 위한 특별한 concurrency control이 필요하게 된다.

이러한 concurrency control을 위하여 이미 제안된 기존의 알고리즘들은 lock을 풀어주기 위해 transaction ID값을 비교하는 방법을 사용하고 있다. 여기서 transaction ID는 관리자와 피관리자 사이에 이루어지는 일련의 구체적인 동작을 정의하며 관리가 수행된 시작 시간과 lock을 요구한 관리자 등이 포함된다. 이 방식을 바탕으로 운용되는 알고리즘으로는 wait-die 알고리즘, running-priority 알고리즘, wound-wait 알고리즘 등이 있으며 이 중에서 wound-wait 알고리즘이 상대적으로 우수한 것으로 평가되어 있다[6].

Wound-wait 알고리즘의 핵심은 피관리자에서 transaction ID값을 비교하여 그 결과를 관리자들에게 전달해 주는 것을 기본으로 하며 이 알고리즘의 절차는

그림 3과 같다.

다른 알고리즘과 비교하였을 때 wound-wait 알고리즘이 가지고 있는 상대적인 장점은 재시작(restart) 횟수가 없다는 점이다. 이는 관리자원 획득에 실패한 관리자가 같은 관리행위를 수행하려고 다시 시도하지 않는다는 것을 의미하며 이는 관리의 효율성 측면에서 아주 중요한 사항이 된다. 그러나 wound-wait 알고리즘은 deadlock을 풀어주는데 필요한 시간이 상대적으로 많은 것으로 나타나며 이는 망과 관리자의 성능이 향상됨에 따라서 치명적인 단점으로 부각된다. 특히 실시간 처리를 요하는 관리체계에서는 시간지연요소가 가장 민감한 사항이 되므로 이러한 단점은 개선되어야 한다. 따라서 이 알고리즘이 가지고 있는 재시작 횟수가 없다는 장점을 유지하면서 시간지연 요소를 개선할 수 있는 새로운 알고리즘이 요구된다.

### III. 새로운 Concurrency Control 방안의 제안

이 연구에서 제안하는 concurrency control 알고리즘의 핵심은 TMN에서의 DCN(Data Communication Network)을 통해 관리자들간에 네트워크가 구성되어 있

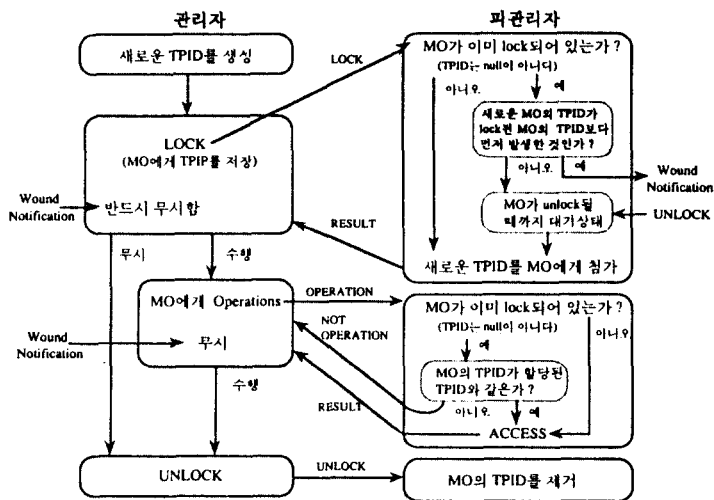


그림 3. Wound-Wait 알고리즘

는 점을 충분히 활용하여 관리자들이 concurrency control에 필요한 사항들에 직접 관여하자는 것이다. 이러한 의미에서 제안한 알고리즘을 관리자간 협상 알고리즘(IMNA: InterManager Negotiation Algorithm)이라고 칭한다. 그림 4는 제안한 관리자간 협상 알고리즘의 동작 절차를 나타낸 것이다.

Wound-wait 알고리즘과 관리자간 협상 알고리즘을 구체적으로 비교하여 보면 다음과 같다. 먼저 wound-wait 알고리즘의 경우 발생된 transaction ID값은 피관리자에서 비교되며 그 결과가 관리자에 통보된 다음 어느 관리자가 피관리자내의 관리자원에 대한 사용권리를 얻을 것인지를 결정하는 간접비교방식을 채택하고 있다. 그러나 관리자간 협상 알고리즘은 transaction ID Generation Notification을 DCN을 통해 다른 관리자들에게 직접 전송하고, 이를 이용하여 관리자들이 자신의 transaction ID 값과 다른 관리자들의 transaction ID 값을 직접 비교하는 방식을 채택하고 있다. 이 비교 결과에 의해 어느 관리자가 관리자원의 사용권리를 가질 것인가가 결정됨에 따라 deadlock 발생시 이를 해결하기 위해 사용되는 메시지의 수가 감소하고 최종 lock 확인까지 걸리는 시간이 단축된다. 그리고, 관리자원의 사용권 획득에 실패한 관리자는 그 시간만큼 다

른 관리에 참여하는 시간이 빨라지게 되고, 피관리자에게는 concurrency control을 위한 부가적인 기능이 없어지게 되므로 기존의 알고리즘들을 적용할 때 받던 부담을 줄이게 된다.

기존의 알고리즘들과 관리자간 협상 알고리즘의 deadlock을 제거하는 절차 및 재시도 횟수 측면을 비교하기 위하여 그림 2의 관리자 A가 관리자 B보다 먼저 lock을 요구하나 deadlock이 발생한 경우를 가정하여 이를 해소하기 위한 각 방식별 절차를 그림 5에 보였다. Wound-wait 알고리즘은 그림 5.a와 같이 피관리자 2가 관리자 A로부터 lock요구를 수신하여 관리자 B에게 wound notification을 통지한다. 그리고 관리자 A에 대한 lock을 설정하여 test를 수행한 후 연속하여 관리자 B에 대한 lock이 이루어지게 되기 때문에 lock요구에 대한 재시도는 발생하지 않는다. 관리자간 협상 알고리즘은 그림 5.b와 같이 NE에서의 wound notification 대신에 OS에서의 LOCK-Confirm을 사용한다. 그리고 관리자 A에 대한 lock을 설정하여 test를 수행한 후 관리자 B에 대한 lock이 이루어지게 되기 때문에 wound-wait 알고리즘과 같이 lock요구에 대한 재시도는 발생하지 않는다. 그러나 running-priority 알고리즘은 새로운 lock요구를 수신할 경우 이전의

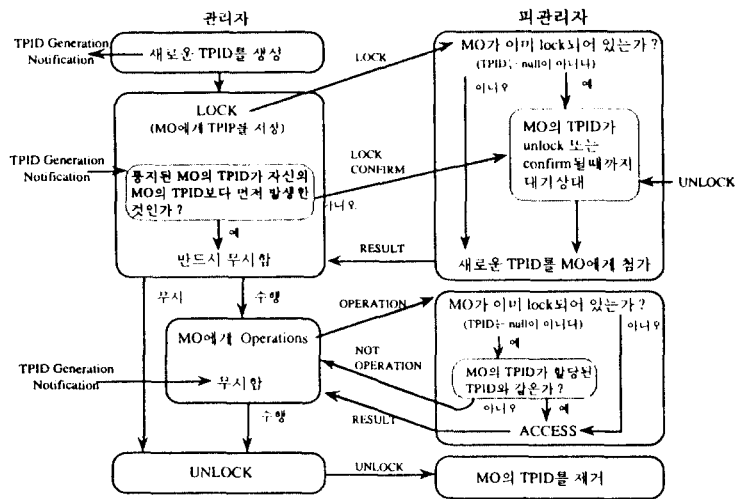


그림 4. 관리자간 협상 알고리즘

lock 시간과는 무관하게 이전의 lock을 해제한 후 새로운 lock 요구를 채택하기 때문에, 그림 5.c와 같이 피관리자 2에서는 관리자 B로 wound notification을 통지하여 관리자 B에 대한 lock을 해제한 후 관리자 A에 대한 lock을 채택하고 피관리자 1에서는 관리자 A로 wound notification을 통지하여 관리자 A에 대한 lock을 해제한 후 관리자 B에 대한 lock을 채택한다. 따라서 관리자 A와 B에 대한 lock이 재시도 되어진다. Wait-die알고리즘은 그림 5.d와 같이 피관리자 1이 관리자 A에 대한 lock을 채택하기 위하여 관리자 B에게 die notification을 통지하여 관리자 B에 대한 lock을

해제하기 때문에 관리자 B에 의한 lock요구가 재시도 되어진다.

이와 같은 동작상의 상이함을 기초로하여 wound-wait알고리즘과 관리자간 협상 알고리즘을 대상으로 구성적 측면의 차이를 비교하면 표 1과 같다.

#### IV. 동기식 전송 관리망에서의 시험관리의 적용

##### 1. 동기식 전송망의 시험관리

동기식 전송망의 시험은 설정된 전송로의 구성 상태나

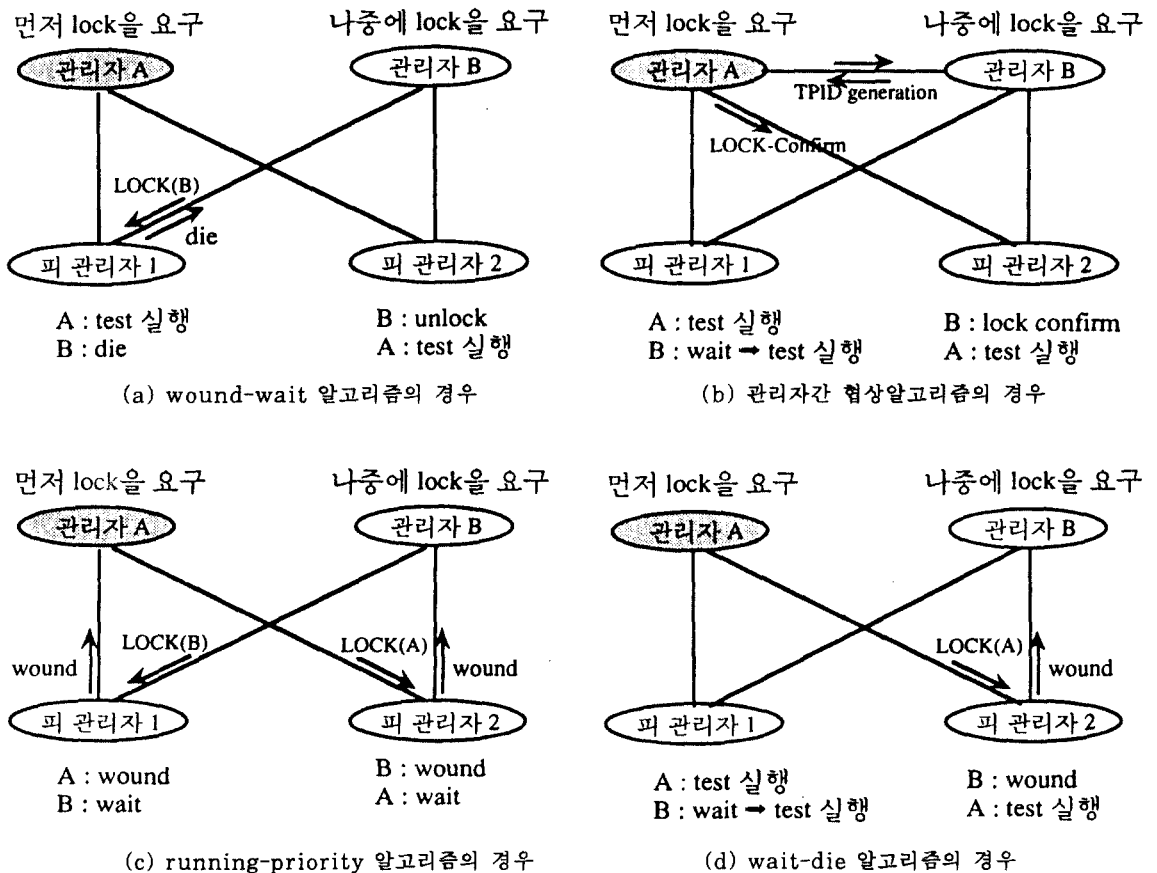


그림 5. Deadlock을 제거하기 위한 절차

성능들을 평가하고, 그 결과값을 이용하여 QOS(Quality Of Service)를 유지하기 위함이다. 이를 위해 OSI의 시험관리기능을 사용하여 STM-N 프레임의 모든 계위수준(hierarchical level)의 전송로를 시험하게 된다.

시험의 종류로는 in service scheme과 out-of-service scheme의 두 가지가 있다. In service scheme은 일상적인 시험 방법으로서 사용자 정보를 차단하지 않고 SOH (Section OverHead)내의 BIP(Bit Interleaved Parity)를 이용하여 시험한다. Out-of-service scheme은 전송로 상에 심각한 장애가 발생하였을 때의 방법으로서 그림 5와 같이 inser-

tion point에서 사용자 정보를 차단한 후 payload에 시험 신호인 PRBS(Pseudo Random Bit Sequence)를 삽입하여 시험구간을 통해 프레임을 전송하고 drop point에서 전송된 시험 프레임을 분석하여 문제가 되는 전송구간의 성능과 상태 등을 평가하는 것이다. 성능과 상태를 분석하는 감시점(monitoring point)은 중간 지점에 위치하여 시험 프레임의 오버헤드를 처리 및 평가한 후 drop point로 시험 프레임을 전송한다.

시험관리를 수행할 때에는 SMF(System Management Function)의 하나인 시험 관리 기능(test management function)을 사용한다[7]. 이는 전송망의 성능을 평가하

표 1. wound-wait 알고리즘과 관리자간 협상알고리즘의 비교

비교요소	wound-wait 알고리즘	관리자간 협상 알고리즘
○관리정보의 전송결로	OS-NE	OS-NE와 OS-OS
○OS의 구성	단순	복잡
○NE의 구성	복잡	단순
○알고리즘이 적용되는 대상	NE	OS
○알고리즘의 절차	비교적 복잡함	비교적 단순함
○재시도 횟수	0	0
○알고리즘의 수행시간	긴 시간이 소요	짧은 시간이 소요
○DCN	필요	필요

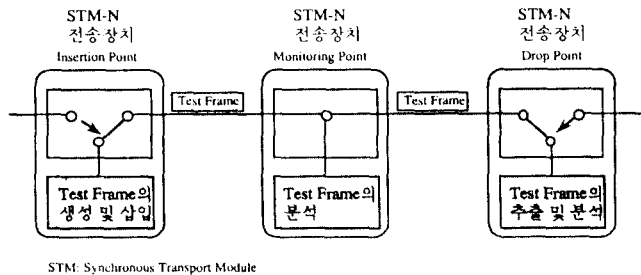


그림 6. Out-of-service scheme에 의한 전송로 시험

거나 장애가 발생하였을 경우 장애 발생의 이유 및 정도를 파악하기 위한 기능을 제공하며, SMFA(Specific Management Function Area) 중 장애관리 영역과 성능관리 영역에서 이 기능을 활용한다[8].

2. Concurrency Control 알고리즘을 적용한 관리망의 구조

그림 6의 모델은 wound-wait 알고리즘을 분산형 동기식 전송 관리망에 적용한 구조로서 시험 관리 정보의 흐름은 관리자와 피관리자 사이의 DCN을 통하여 이루어진다. 이 모델을 이후 type 1이라고 약칭한다.

관리자간 협상 알고리즘을 적용한 분산형 동기식 전송 관리망의 구성 형태도 물리적으로는 앞의 경우와 유사하나 그림 7과 같이 DCN을 관리자들간의 관리정보 전달 경로로서 충분히 활용한 형태이다. 이 모델의 장점은 관리자들간에 관리 정보를 공유함으로써 분산 관리 환경에서 제공되는 기능을 보다 많이 활용할 수 있다는 것이다. 특히 관리 시스템의 컴퓨팅 능력이 급속히 향상됨에 따라 이와 같은 구조는 분산형 관리 환경에 보다 적합한 구조가 될 것이다. 이 모델을 이후 type 2라고 약칭한다.

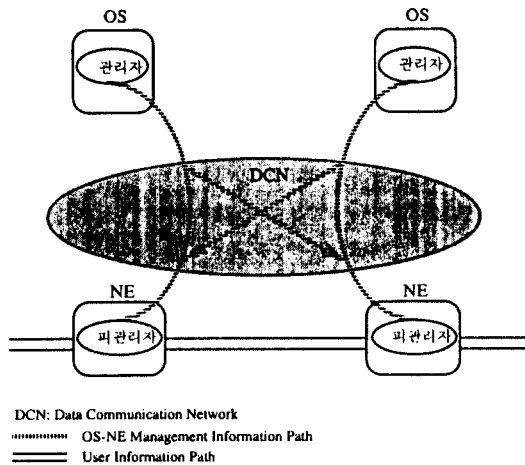


그림 7. Wound-wait 알고리즘을 적용한 관리망의 구조

3. 시험관리 절차

Type 1과 type 2의 실제 전송로 시험관리에서 사용할 수 있는 시험관리 절차를 설정하면 그림 8과 같으며 이 절차는 알고리즘의 종류에 상관없이 공통적으로 적용된다. 시험관리 절차의 전체적인 구성은 크게 locking 단계, 환경 설정 단계, 그리고 시험 수행 단계의 세 가지로 구분된다. 시험관리 절차는 CMIS(Common Management Information Service)의 M-ACTION 서비스를 이용하여 performer에게 controlled test를 요구하고 TO(Test Object)가 생성되면서 시작된다[9].

Locking 단계에서 관리자는 객체관리기능(object management function)이 제공하는 PT-SET 서비스를 이용하여 시험관리와 관련된 TO에 대한 사용권리를 요구한다[10]. 이때 여러 관리자들이 시험관리에 참가하게 될 경우에는 concurrency control 알고리즘을 수행하게 되고, 한 관리자가 관리자원에 대한 사용권리를 모두 획득한 후에 비로소 환경 설정 단계로 넘어간다. 환경설정 단계는 시험관리에 필요한 준비를 하는 것으로 역시 PT-SET 서비스를 이용한다. 실제 전송로 시험의 시작은 시험 수행 단계에서 CMIS의 M-ACTION 서비스를 시험 감시 및 분석을 담당하는 피

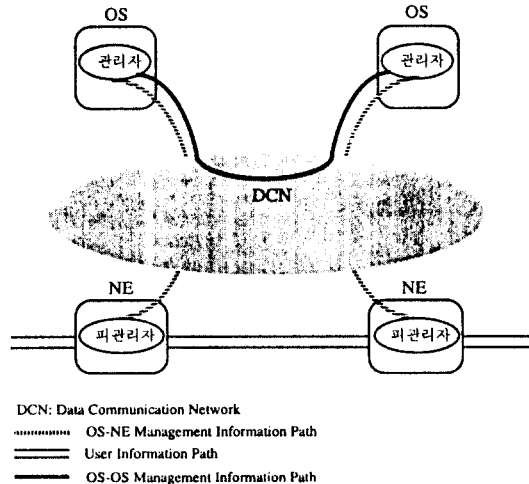


그림 8. 관리자간 협상 알고리즘을 적용한 관리망의 구조



관리자에게 전달하면서 이루어지고, 전송로 시험의 결과는 TO가 notification의 형태로 관리자에게 전달하여 준다. 관리자는 M-ACTION 서비스를 이용하여 시험 관리를 중단시키며 필요할 경우 PT-DELETE 서비스를 이용하여 TO를 소멸시킨다.

### V. 성능분석

#### 1. 시뮬레이션 파라미터값의 설정

Type 1 모델과 type 2 모델을 상대적으로 비교할 수 있도록 다음과 같은 가정으로 시뮬레이션 환경 파라미터값을 설정하였다. 먼저 DCN의 전송속도는 DCC1(Data Communication Channel 1)의 192kbps로 가정하여 관리정보를 전송할 수 있는 통신채널을 구성한다. 장치간의 거리는 20km로 가정하고 여기에 광케이블의 전송특성을 정합시켜서 전송링크상의 전파지연시간(prop-

agation delay)을 약 10msec로 하였다. 이는 실제의 동기식 전송 관리망의 DCN을 따르고 있기 때문에 본 시뮬레이션의 환경을 실제 네트워크와 보다 유사하게 구현하기 위한 것이다.

시험관리의 환경측면에서 시험 프레임 생성 및 추출하고 평가하는데 소요되는 시간을 평균 10 msec의 exponential 분포로 가정하였다. 그리고 시험 요구의 발생 간격은 어느 특정 MO의 관점에서 볼 때 단지 몇 분에서 수일 ~ 수 주일 혹은 수개월의 다양한 분포의 시험요구 간격을 보일 수 있기 때문에 어느 정도 합리적인 값을 산출해 내기가 무척 어렵다. 따라서 본 시뮬레이션에서는 시험 요구 발생 간격을 평균 3600sec의 exponential 분포로 가정하여 수행하였다.

시뮬레이션 수행시 관심 대상이 되는 입력 변수들은 다음과 같이 정하였다.

- OS의 성능(SR<sub>OS</sub>: Service Rate of OS)

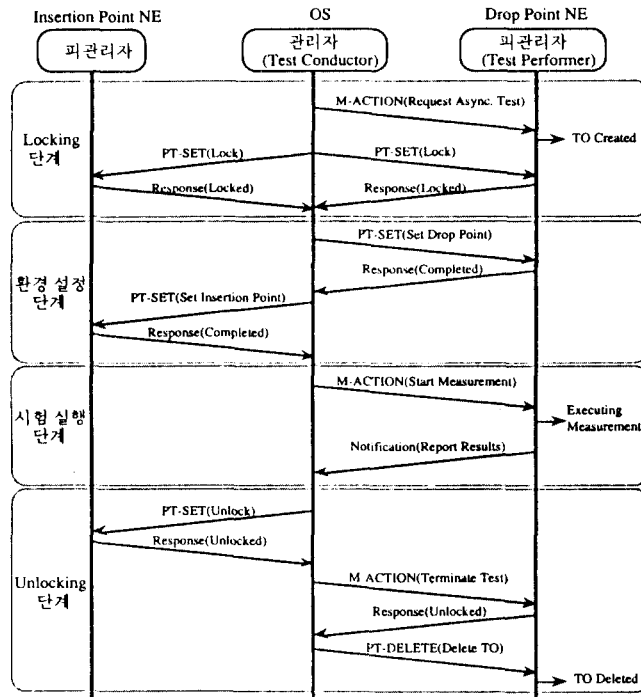


그림 9. 전송로 시험관리 절차

- 평균 시험 요구의 갯수 ( $N_{AR}$ : Average Number of Alarm Report)
- NE의 갯수 ( $N_{NE}$ : Number of NE)

OS의 서비스율에 따른 lock 지연시간을 분석하기 위하여 NE내 피관리자의 평균 서비스율을 100 message/sec라 가정하고 이를 기준으로 하여 OS내 관리자의 평균 서비스율로 100 message/sec ~ 1000 message/sec의 값을 적용하였다. 이는 OS의 평균서비스율은 자신에게 관리를 받는 NE의 갯수에 상당한 영향을 받게되고 또한 하나의 OS가 동시에 여러개의 NE들을 관리할 경우에 대한 영향력을 고려하기 위해서 이다. 그리고 다양한 환경하에서 시험요구가 수행될 때까지의 대기시간을 구하기 위하여 다수의 시험 요구가 동시에 발생하도록 하였으며 1부터 20사이의 평균값을 가지는 poisson 분포로 가정하였다. 한편 OS는 2개를 기본으로 하고 NE의 갯수는 2개 부터 4개로 변화시켰는데  $N_{NE} = 2$ 인 경우는 그림 6과 같이 한개의 전송로시험관리를 위한 insertion point와 drop point가 각각 하나씩 존재하는 형태이고,  $N_{NE} = 4$ 인 경우에는 두개의 전송로 시험관리를 위한 insertion point와 drop point가 각각 두개씩 존재하는 형태가 된다.

시뮬레이션의 결과로 NE와 OS에서의 지연시간을 평가하였는데, NE의 경우에는 시험요구가 발생하여 이 시험관리요구가 완료될 때까지의 지연시간(그림10~13)으로 평가하였고, OS 경우에는 OS가 시험요구를 수신한 후 시험관리를 획득하지 못하고 idle 상태로 복귀할 때 까지의 시간(그림 14) 과 OS가 시험요구를 수신한 후 시험관리를 획득하여 시험 관리를 수행하기 직전까지의 지연시간 (그림 15)으로 평가하였다.

## 2. 시뮬레이션 결과

관리자간 협상 알고리즘은 wound-wait 알고리즘이 기존에 제안된 알고리즘보다 상대적으로 우수하다는 장점인 재시도 횟수가 없다는 특징을 이미 살펴본 바와 같이 동일하게 보유하며, wound-wait 알고리즘과 비교하여 시간지연 요소의 개선된 사항을 시뮬레이션 결과로서 살펴보면 다음과 같다.

OS의 서비스율은 관리자간 협상 알고리즘의 성능을 분석하는 주요 사항으로서 OS의 평균 서비스율(100 message/sec ~ 1000 message/sec)을 NE의 평균 서비스율(100 message/sec)의 배수로 증가시키면서

시뮬레이션을 수행하였으며 그 결과를 그림 10과 그림 11에 나타내었다. 그림 10은 NE의 갯수가 2개일 경우이고 그림 11은 NE의 갯수가 4개인 경우로서 한번에 발생하는 평균 시험요구의 갯수를 5개로 하여 두 경우 ( $N_{NE} = 2, 4$ )에 모두 동일하게 적용하여 시뮬레이션한 결과값이다. 이 결과값은 NE의 피관리자에서 측정된 시간으로서 시험요구가 발생한 시간부터 시험관리요구가 완료될 때까지의 지연시간이다. 시뮬레이션 결과를 살펴보면 그림 10에서는 type1과 type2의 시간지연차가 그리 크지 않으나 그림 11의 type1의 경우에는  $N_{NE}$ 이 2에서 4로 증가함에 따라 시간지연값이 거의 2배로 증가하고 type2의 경우에는 시간지연차가 거의 발생하지 않음을 볼 수 있다. 이는 관리자간 협상알고리즘이 NE에 산재되어 있는 부하를 OS에게 전이시키므로서 NE의 갯수가 증가함에 따라 발생할 수 있는 시간지연요소를 줄일 수 있음을 나타낸다.

관리 환경 측면에서 평균 시험 요구의 갯수는 다양하게 변하는 관리 요구를 나타내는 요소로서 1개에서 19개까지 변화시키면서 시뮬레이션을 수행하였다. 그림 12는 NE의 갯수가 2개이고 그림 13은 NE의 갯수가 4개인 경우로서 OS의 서비스율에 의해 wound-wait 알고리즘이나 관리자간 협상알고리즘이 영향받지 않도록 정상상태라고 추정되는 OS서비스율 100 message/sec로 하여 시뮬레이션한 결과이나 그림 10과 11에서 예측할 수 있듯이 서비스율이 500 message/sec 이상이면 큰 차이가 없을 것으로 예상된다. 이 결과값은 NE의 피관리자에서 측정된 시간으로서 시험요구가 발생한 시간부터 시험관리요구가 완료될 때까지의 지연시간이다. 그림 12에서는 type1과 type2의 시간지연차가 크지 않으나 그림 13에서는  $N_{NE}$ 이 2에서 4로 증가할 경우에 평균 시험요구갯수가 증가하면서 type1과 type2의 시간지연차가 점점 커짐을 볼 수 있다. 이는 다양하게 변하는 관리요구에 대처하는 능력이 기존의 알고리즘보다 본 연구에서 제안한 관리자 협상 알고리즘이 우수함을 나타낸다.

그림 14과 그림 15는 관리자에서의 시험관리능력을 살펴보기 위해서 NE를 2개로, 평균 시험요구갯수를 5개로 고정시키고 OS의 서비스율을 100 ~ 1000 메시지/초로 변화시키면서 시뮬레이션을 수행한 결과이다. 그림 14은 quit delay를 비교치로 설정하였는데, 이는 OS의 관리자에서 시험관리 요구를 수신한 시간부터 전송로 시험 수행권리를 획득하지 못하고 idle 상태로 복

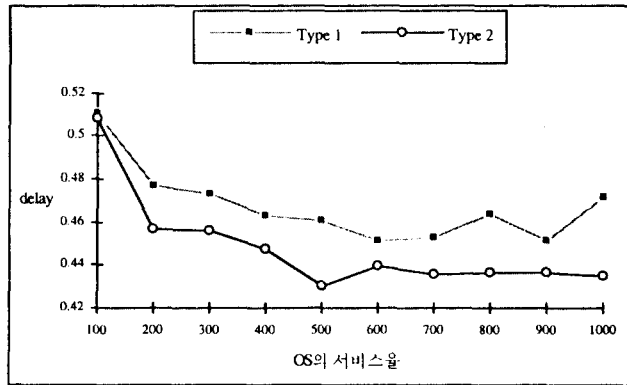


그림 10. OS의 서비스율의 변화에 따른 NE에서의 지연시간  
( $N_{NE} = 2, AN_{AR} = 5$ )

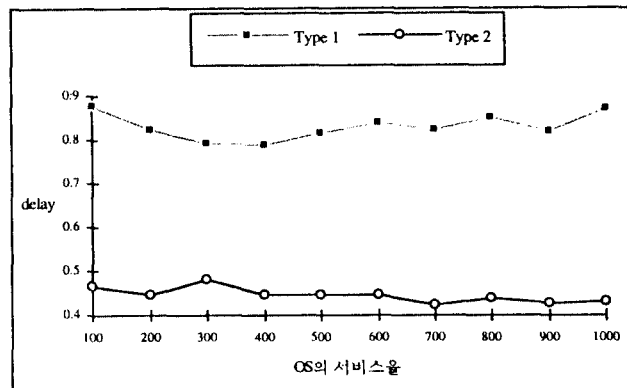


그림 11. OS의 서비스율의 변화에 따른 NE에서의 지연시간  
( $N_{NE} = 4, AN_{AR} = 5$ )

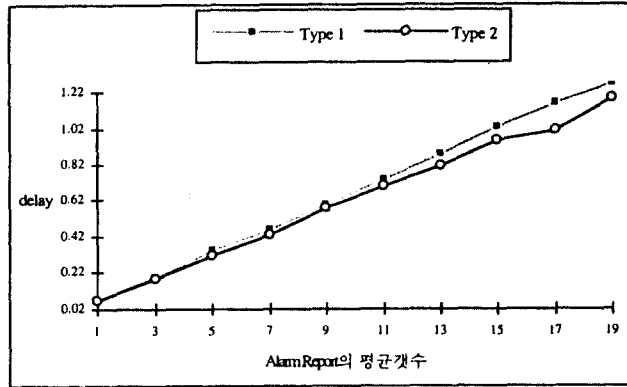


그림 12. 평균 TO의 갯수 변화에 따른 NE에서의 지연시간  
( $N_{NB} = 2, SR_{CS} = 1000$  메시지/초)

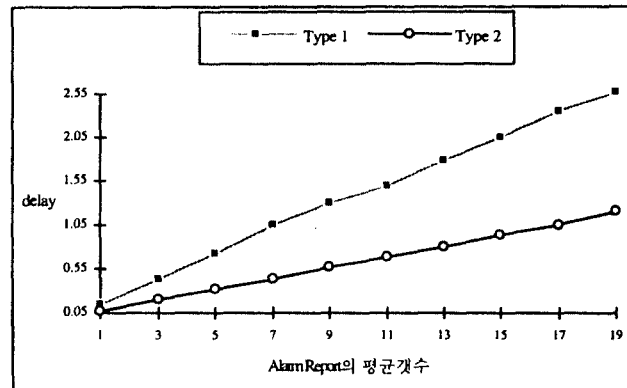


그림 13. 평균 TO의 갯수 변화에 따른 NE에서의 지연시간  
( $N_{NB} = 4, SR_{CS} = 1000$  메시지/초)

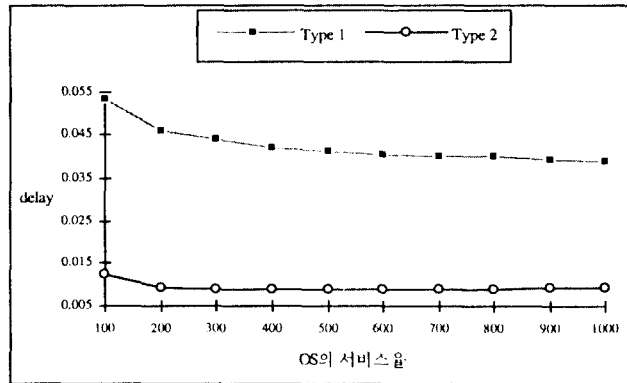


그림 14. OS의 서비스율에 따른 OS에서의 quit delay  
( $N_{NE} = 2, AV_{AR} = 5$ )

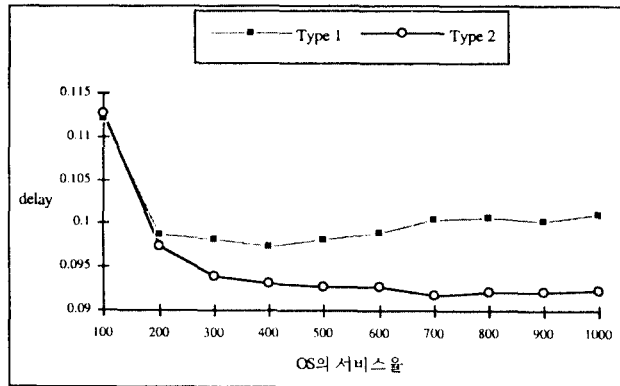


그림 15. OS의 서비스율에 따른 OS에서의 test start delay  
( $N_{NE} = 2, AV_{AR} = 5$ )

귀할 때까지의 시간이다. 이 수행 결과값은 시험관리권을 획득하지 못한 관리자가 원래의 상태로 복귀하는 시간이 기존의 알고리즘보다 제안한 알고리즘이 짧다는 것을 입증한다. 그림 15는 test start delay를 비교치로 설정하였는데, 이는 OS의 관리자에서 시험관리요구를 수신한 시간부터 시험수행권리를 획득하여 test를 수행하기 직전까지의 시간이다. 결과값을 보면 OS의 성능이 좋아질 수록 type 2의 성능이 type 1의 성능보다 우수한 것을 알 수 있으며 결론적으로 관리자가 시험관리권을 획득할 때까지의 시간이 기존의 알고리즘보다 제안한 알고리즘이 짧다는 것을 입증하는 것이다.

## VI. 결 론

본 연구에서는 분산 관리 환경에서 관리 수행시 발생할 수 있는 문제점인 deadlock에 대해 중점적으로 살펴보고 이를 해결하기 위해 적용할 수 있는 concurrency control 알고리즘들을 살펴보았다. 그리고 기존의 concurrency control 알고리즘들의 특성과 문제점을 분석하고, 이를 해결하기 위한 새로운 concurrency control 알고리즘인 관리자간 협상 알고리즘을 제안하였으며 그 성능분석을 위하여 제안한 알고리즘과 기존의 알고리즘을 분산형 동기식 전송망의 전송로 시험관리에 적용시켜 시뮬레이션을 수행하였다.

관리자의 성능이 향상되거나 망의 구조가 복잡해질수록 관리자간 협상 알고리즘을 적용한 분산형 동기식 전송 관리망이 전송로 시험관리 수행에 필요한 시간을 최소화시키고 망의 자원을 충분히 활용하여 효율적인 관리 환경을 구축할 수 있다는 것을 시뮬레이션 결과로서 입증하였다.

결론적으로 관리자간 협상 알고리즘이 분산형 동기식 전송 관리망에서의 시험관리와 같이 분산 관리망을 구현할 때 발생하는 여러가지 제반 문제들을 해결할 수 있는

알고리즘이 될 것으로 기대한다.

## 참고문헌

1. ITU-T Rec. G.784 SDH Management(1990).
2. ITU-T Rec. M.3010 Principles for a Telecommunications Management Network(1992).
3. P.J.Brusil, W. Collins, "OSI Systems Management Standards," GLOBECOM '91.
4. G. A. Champine, "Distributed Computer Systems," North-Holland Publishing Company, 1980.
5. R. Agrawal, M. J. Carey, L. W. McVoy, "The Performance of Alternative Strategies for Dealing with Deadlocks in Database Management Systems," IEEE Trans. Software Eng., vol. SE-13, no. 12, Dec. 1987.
6. S. Ohta and N. Fujii, "Applying OSI Systems Management Standards to Remotely Controlled Virtual Path Testing in ATM Networks," IEICE Trans. Comm., vol. E76- B, no. 3, Mar. 1993.
7. ITU-T Draft Rec. X.745 Test Management Function, 1993.
8. W. Stallings, "SNMP, SNMPv2, and CMIP," Addison-Wesley Publishing Co., 1993.
9. ITU-T Rec. X.710 Common Management Information Service Definition, 1991.
10. ITU-T Draft Rec. X.730 Object Management Function, 1992.

李 玟 炯(Min Hyoung Lee)

정회원

1969년 5월 23일생

1993년 2월 : 성균관대학교 전자공학과 공학사  
1995년 2월 : 성균관대학교 대학원 전자공학과 공학석사  
1995년 3월~현재 : 대우 통신 연구소 연구원



朴 明 桓(Myeong Hwan Park) 정회원

1965년 9월 20일생

1991년 2월 : 성균관대학교 전자공학과 공학사  
1994년 2월 : 성균관대학교 대학원 전자공학과 공학석사  
1994년 3월~현재 : 성균관대학교 대학원 전자공학과 박사과정 재학중



宋 重 求(Joog Goo Song) 정회원

1957년 5월 21일생

1979년 2월 : 성균관대학교 전자공학과 공학사  
1988년 2월 : 성균관대학교 대학원 전자공학과 공학석사  
1992년 3월~현재 : 성균관대학교 대학원 전자공학과 박사과정 재학중



曹 圭 燮(Kyu Seob Cho) 정회원

1951년 5월 3일생

1974년 2월 : 성균관대학교 전자공학과 공학사  
1976년 2월 : 성균관대학교 대학원 전기공학과 공학석사  
1989년 2월 : 성균관대학교 대학원 전자공학과 공학박사  
1977년 3월~1992년 2월 : ETRI 책임연구원  
1993년 3월~현재 : 성균관대학교 전자공학과 부교수

朴 炳 哲(Byung Chul Park)

정회원

1930년 4월 30일생

1957년 9월 : 서울대학교 통신공학과 공학사  
1975년 2월 : 인하대학교 대학원 전기공학과 공학박사  
1980년 9월 : 일본 동경대학 외국인연구원(1년간)  
1972년 3월~현재 : 성균관대학교 전자공학과 교수