

논문 2010-47TC-2-5

이중 OS로 구동되는 Dual CPU 기반에서의 S/W Watch-Dog 기법

(S/W Watch-Dog method between dual CPU using different OS)

유 영 일*, 전 병 실**

(Young-Eel You and Byoung-Sil Chon)

요 약

네트워크 장비의 데이터 처리 속도와 용량이 급속하게 향상되면서 기존 단일 프로세서로는 데이터 처리 성능에 한계에 이르고 있다. 즉, 하나의 CPU에서 Unit에 대한 Management 기능에서 Data Processing 등 모든 기능을 수행할 경우 데이터 처리 속도와 용량에 제약이 따르게 된다. 이를 보완하기 위해서 하나의 프로세서는 Unit Management 기능만을 처리하고 다른 하나 이상의 프로세서에서는 Data Processing 관련 기능을 전담하여 시스템의 안정성과 많은 데이터를 보다 빠른 시간에 처리할 수 있는 신뢰성에 초점을 맞추고 있다. 본 논문에서는 이중 OS로 구동되고 있는 Dual CPU 기반에서의 S/W Watch-Dog 기능을 제안한다. 즉, 상이한 OS를 사용하는 2개의 프로세서로 구성된 Unit에서 프로세서간 통신 장애나 프로세서 Down 등을 감시/감지하는 방법에 대해 제안하고, 제안한 방법에 대해 S/W Watch-Dog 기능의 신뢰성을 극대화 할 수 있는 요소(임계 시간, 우선순위)의 최적 값을 시험 및 분석 Tool을 사용하여 찾아내고, Data Rate에 따른 S/W Watch-Dog 기능의 정확성을 향상시킬 수 있다.

Abstract

This paper proposes S/W Watch-Dog method between Dual CPU using the different OS. The proposed watch-dog method performs that it distinguishes the status of channel between dual CPU and status of processor itself. We find out the ideal value of threshold and priority for load test task, and we evaluate the accuracy of the proposed S/W Watch-Dog Method at the result of evaluation. We figure out that the accuracy of proposed method is higher than the accuracy of general S/W Watch-Dog Method in case of variable data rate. Therefore we confirm that the proposed Method has high accuracy of watch-dog function with the ideal value of threshold and priority for load test task through the performance evaluation.

Keywords : 5Watch-Dog, OS, Threshold, Priority

I. 서 론

최근에는 기능의 다양화/집적화, 데이터 처리 용량 및 속도를 향상시키기 위해 하나의 Processor는 Unit에 대한 Management 기능만을 수행하고, 다른 한 개 이상의 프로세서들은 Data Processing 관련 기능을 전담하는 구조로 System이 변모하고 있다. 이처럼 다수의

Processor를 사용하게 되면서 각각의 Processor들에 대한 상태 및 장애 정도를 하나의 Processor에서 처리하여 전체적으로는 유기적인 관리의 필요성이 대두되고 있다. 이를 위한 하나의 방법이 S/W Watch-Dog 기능을 사용하는 것인데, S/W Watch-Dog 기능은 심각한 기능 상의 문제 또는 불능 상태에 빠져 있는 System/Unit 등을 강제로 Restart 시키는 역할을 수행하여, 시스템이 정상 상태로 천이되도록 한다.

본 논문에서는 상이한 OS로 구동되고 있는 Dual기반에서의 S/W Watch-Dog 기능을 제안한다. 즉, 이중 OS를 사용하는 2개의 프로세서로 구성된 Unit에서 프로세서간 통신 채널 활성화 여부 및 통신 상태 장애 여

* 정희원, 서울통신기술(주) 통신시스템 연구소
(SEOUL COMMTECH CO., LTD.)

** 정희원, 전북대학교 공과대학 전자정보공학부
(Division of Electronic and Information Eng.,
Chonbuk Nat'l Univ.)

접수일자: 2008년10월24일, 수정완료일: 2010년2월17일

부를 판단하고, 프로세서 자신의 불능 상태 등을 감지/감지하는 방법에 대해 기술하고, 제안된 기법에서 Watch-Dog 기능을 수행하는 Task의 임계 시간 (Threshold) 및 우선 순위(Priority)의 최적 값을 시험 및 분석 Tool을 사용하여 찾아내어, Data Rate의 변화에 따른 S/W Watch-Dog 기능의 정확성을 극대화 시킬 수 있었다.기에 서론을 입력하세요.

II. 본 론

1. 기존 단일 프로세서 기반의 Watch-dog 기법

일반적인 S/W Watch-Dog 기능은 그림 1에서 보는 바와 같이 Processor와 Boot에서 하나의 전역 변수 (Watch-Dog Counter)를 이용하여 프로세서의 이상 유무를 판단한다. 즉 Processor에서는 1초 주기로 이 Watch-Dog Counter(Processor가 불능 상태에 빠져 재시동되기 전까지의 최대 시간 값을 의미) 값을 일정한 시간의 값으로 설정하고 Boot에서는 이 Counter 값을 지속적으로 감소시킨다. 이때 프로세서에서 이 Watch-Dog Count를 일정한 시간 값으로 설정하지 못하는 장애가 발생하거나 불능 상태에 이르게 되면, Boot에서는 이 시간 값을 지속적으로 감소시키며 이 시간의 값이 0(영)에 다다르게 되는 순간 Boot에서는 해당 Processor에 장애가 발생함을 감지하고 Processor를 Restart시킴으로써 시스템이 정상 상태로 천이되도록 한다.

최근에는 기능의 다양화/집적화, 데이터 처리 용량 및 속도를 향상시키기 위해 하나의 Processor는 Unit에 대한 Management 기능만을 수행하고, 다른 한 개 이상의 프로세서들은 Data Processing 관련 기능을 전담하는 구조로 변모하고 있다. 그래서 기존 단일 Processor 기반 위에 고성능의 Network Processor를 장착하여 시스템을 최적화하고 있으나, 각각의 Processor가 상이한

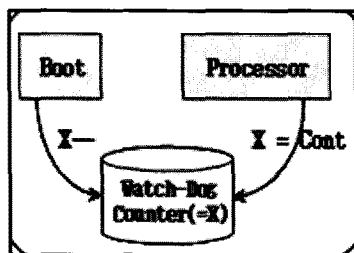


그림 1. 일반적인 S/W Watch-Dog 과정
Fig. 1. General S/W Watch-Dog Flow.

를 사용하여 구동되는 시스템의 경우, 그림 1과 같은 S/W Watch-Dog 기법을 프로세서마다 적용할 경우 기능의 중복 구현이 발생하며, 각 프로세서간 개별적인 Watch-Dog 동작으로 인해 Management 기능과 Data Processing 기능에 악영향을 미치게 된다.

2. Dual CPU 구조에서 효과적인 S/W

Watch-Dog 기능 제공 방안 및 설계

일반적인 S/W Watch-Dog 기법은 상이한 OS를 사용하는 Dual CPU 구조에서 신뢰성 있는 Watch-Dog 기능이 불가능하다.

본 논문에서는 상이한 OS를 사용하는 Dual CPU 기반에서 S/W Watch-Dog 기능의 신뢰성을 극대화 할 수 있는 방안을 제안하고, 제안된 기법에서 Watch-Dog 기능을 수행하는 Task의 임계 시간 및 우선 순위의 최적 값을 시험 및 분석 Tool을 사용하여 찾아내어, Data Rate에 따른 S/W Watch-Dog 기능의 정확성을 극대화 시키고 있다.

제안된 기법에서는 그림 2에서 보는 바와 같이 Unit 가 재시동을 하게 되면 가장 먼저 Boot가 구동되고 Master Processor와 함께 공유하는 Watch-Dog

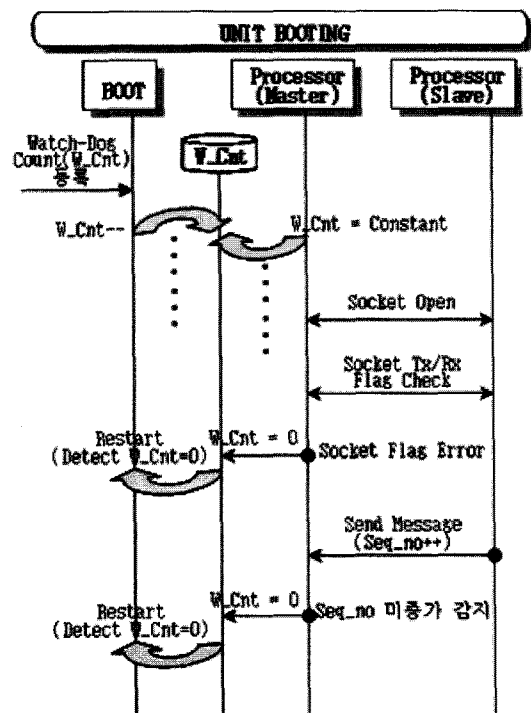


그림 2. 상이한 OS를 사용하는 Dual CPU 기반에서 제안된 S/W Watch-Dog 기법
Fig. 2. Proposed S/W Watch-Dog method for Dual CPU using different OS.

OS

Counter(Processor가 불능 상태에 빠져 재시동되기 전까지의 최대 시간 값을 의미) 값을 일정한 시간의 값으로 설정한다. 이후 Management 기능을 전담하는 Master Processor는 Watch-Dog 기능을 수행하는 Task를 구동시키며, Watch-Dog 기능을 수행하는 Task는 1초 주기로 Watch-Dog Counter 값을 일정한 값으로 등록하면서 동시에 BOOT는 Watch-Dog Counter 값을 계속해서 감소시키는 기능을 수행하게 되는데, 이 값이 0(영) 이하가 되는 시점을 감지하고 Unit을 Restart시키게 된다.

또한 Master Processor는 Data 처리를 담당하는 Slave Processor와 통신 채널(Socket)을 설정하고 설정된 통신 채널이 일정 시간 내에 활성화되는지를 Socket TX/RX 플래그 값으로 판별하는데, 이 플래그 값이 시간 내에 Active되지 않으면 Slave Processor에 장애가 발생한 것으로 판단하여 Watch-Dog Counter 값을 0(영)으로 등록한다. 이때 BOOT는 Watch-Dog Counter 값이 0(영) 이하가 되는 시점을 감지하고 Unit을 Restart시키게 된다. 그리고 Master Processor와 Slave Processor간 통신 채널이 활성화가 된 후, Slave Processor는 일정 주기로 Sequence 값을 증가시키면서 Master Processor에게 Message를 보내게 되고, Master Process는 수신된 메시지 내의 Sequence 값이 증가되는지를 확인한다. 만약 수신된 Message의 Sequence 값이 설정된 Guard-Time 동안 증가되지 않거나 일정 시간 동안 Slave Processor로부터 수신되는 Message 없을 경우, Master Processor는 Slave Processor에 장애가 발생한 것으로 판단하여 Watch-Dog Counter 값을 0(영)으로 등록한다. 이때 BOOT는 Watch-Dog Counter 값이 0(영) 이하가 되는 시점을 감지하고 Unit을 Restart시키게 된다. 이 과정은 한 개 이상의 Processor에서 장애가 인지되기 전까지 반복 수행을 하게 된다.

제안된 기법에서는 상이한 OS를 사용하는 Dual CPU 기반에서 각각의 Processor들이 독립적인 Watch-Dog 기능을 수행하는 것을 방지하고, Master 역할을 하는 Processor에서 다른 Processor의 장애 상태를 판단하여 Unit 단위의 Restart를 통해 Management 기능과 Data 처리 기능의 동기화가 이루어지게 되고, 보다 안정적인 시스템 운용이 가능하도록 한다.

III. 실험

1. 성능 평가를 위한 모델링

본 논문에서 그림 3과 같이 성능 평가를 위하여 Watch-Dog 상황을 유발할 수 있는 시험용 Task를 생성하고, 이 Task의 우선 순위(Priority)와 임계 시간(Task의 동작 지속 시간)을 변경하여 식(1)의 결과 값이 1에 가까운 최적의 Task와 임계 시간을 측정하였다. 또한 10개의 E1 링크를 통해 Data Rate를 10Mbps ~ 20Mbps까지 변화시키면서 기존의 S/W Watch-Dog 기법과 제안된 Watch-Dog 기법에 따른 각각의 Processor의 부하를 측정하였다.

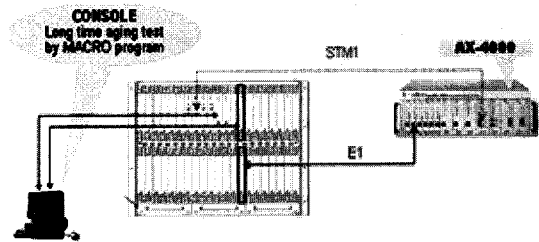


그림 3. 성능 평가를 위한 네트워크 모델링
Fig. 3. Network Modeling for Performance Evaluation.

$$1 - (1[N]/1000[T]) \tag{1}$$

- N : watchdog 기능이 실패 한 횟수
- T : Test Task 실행 횟수

2. Test Task의 임계 시간과 Watch-Dog

표 1은 Watch-Dog 상황을 유발하는 Test Task의 구동 시간(임계 시간)의 변화에 따른 Watch-Dog 기능의 성공 횟수를 나타내고 있다. 표 1과 같이 임계 시간이 7초 인 경우 식(1)의 결과 값이 1이 되며, 이 임계 시간 값(7 sec)이 Watch-Dog 기능의 최적 임계 값이 된다.

표 1. Test Task의 임계 시간 변화에 따른 Watch-Dog 기능의 성공 횟수

Table 1. Watch-Dog success count for change of Test Task threshold.

| 임계시간 (sec) / Count | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------------------------|---|----|----|----|----|----|----|
| Watch-Dog 성공 횟수 (20회 수행) | 5 | 15 | 17 | 17 | 20 | 19 | 19 |

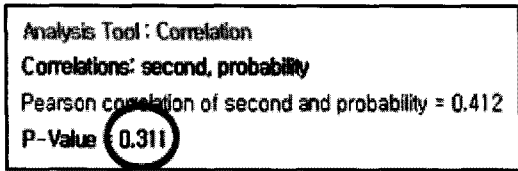


그림 4. Test Task의 임계 시간과 Watch-Dog 기능의 상관 관계
 Fig. 4. Correlation of test task threshold and Watch-Dog Function.

또한 Test Task의 임계 시간 값과 Watch-Dog 기능의 상관 관계를 분석한 결과는 그림 4에서는 보는 바와 같다. 그림 4에서 Sample Size는 Case별 20회 측정 값, X 인자로 Test Task의 임계 시간을 사용하여 상관 분석(Correlation)한 결과 상관 계수(r)는 $r = 0.837$, P-Value = 0.038 를 확인하게 된다. 이는 Test Task의 임계 시간이 Watch-Dog 기능에 영향을 미치고 있음을 나타내고, 임계 시간의 최적 값은 표 1에 보이는 바와 같이 7[sec]임을 확인 할 수 있다.

3. Test Task의 Priority와 Watch-Dog

표 2는 Watch-Dog 상황을 유발하는 Test Task의 우선 순위(Priority)의 변화에 따른 Watch-Dog 기능의 성공 횟수를 나타내고 있다. 표 2와 같이 Test Task의 Priority 값이 127인 경우 식(1)의 결과 값이 1이 되며, 이 Priority 값(127)이 Watch-Dog 기능의 최적 Priority

표 2. Test Task의 Priority 변화에 따른 Watch-Dog 기능의 성공 횟수
 Table 2. Watch-Dog success count for change of Test Task priority.

| Priority | 144 | 160 | 192 | 200 | 225 |
|--------------------------|-----|-----|-----|-----|-----|
| Count | | | | | |
| Watch-Dog 성공 횟수 (20회 수행) | 10 | 8 | 8 | 9 | 7 |

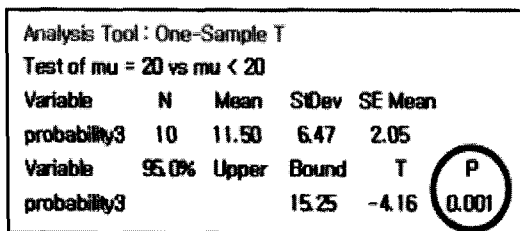


그림 5. Test Task의 Priority 값과 Watch-Dog 기능의 상관 관계
 Fig. 5. Correlation of test task priority and Watch-Dog function.

값이 된다.

또한 Test Task의 Priority 값과 Watch-Dog 기능의 상관 관계를 분석한 결과는 그림 5에서 보는 바와 같다. 그림 5에서 Sample Size는 Case별 20회 측정 값, X 인자로 Test Task의 Priority를 사용하여 One-Sample T 분석 결과 P = 0.001 를 확인하게 된다. P = 0.001은 유의 수준 값 0.005 보다 작기 때문에 Test Task의 Priority 값이 Watch-Dog 기능에 영향을 미치고 있음을 알 수 있고, Test Task의 최적의 Priority 값은 표 2와 같이 127임을 확인 할 수 있다.

4. Test Task의 임계 시간/우선 순위와 Watch-Dog

Test Task의 임계 시간/우선순위와 Watch-Dog 기능과의 영향성에 대해 각각 확인하였다. 본 절에서는 Test Task의 임계 시간과 우선순위를 동시에 변화시킬 경우 Watch-Dog 기능에 대한 영향성을 확인한다.

표 3에서 보는 바와 같이 Test Task의 Threshold (임계 시간)값이 7[sec]이고 Priority 값이 127 인 경우, Watch-Dog Completion 값이 1,000 됨을 알 수 있다. 여기서 Watch-Dog Completion은 Load Test Task를 1,000회 실행했을 때, Watch-Dog 기능이 구동된 횟수를 의미하고, Threshold = 7, Priority = 127인 Test Task 구동시 최적의 Watch-Dog 기능을 수행 할 수 있다.

Watch-Dog 기능의 신뢰성 극대화를 위해 DOE 최적화 분석 결과 그림 6에서 보는 바와 같이 Test Task의 Threshold = 7, Priority = 127 일 때 Watch-Dog 기

표 3. Test Task의 임계 시간과 우선순위 변화에 따른 Watch-Dog 기능 성공 횟수
 Table 3. Watch-Dog success count for change of test task priority and threshold.

| NO | THRESHOLD | PRIORITY | WATCH-DOG COMPLETION |
|----|-----------|----------|----------------------|
| 1 | 7 | 144 | 862 |
| 2 | 7 | 144 | 815 |
| 3 | 3 | 127 | 674 |
| 4 | 3 | 144 | 539 |
| 5 | 7 | 127 | 1000 |
| 6 | 7 | 127 | 998 |
| 7 | 3 | 144 | 596 |
| 8 | 3 | 144 | 618 |

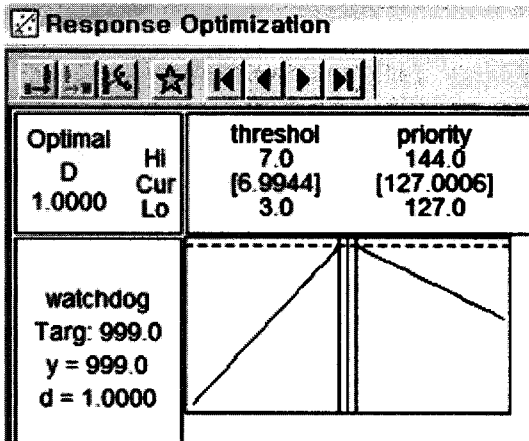


그림 6. Watch-Dog 기능 최적화
 Fig. 6. Watch-Dog Function Optimization.

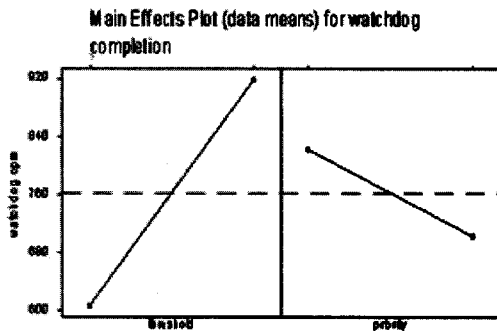


그림 7. Test Task의 임계 시간과 Priority 변화에 따른 Watch-Dog 기능의 영향.
 Fig. 7. Watch-Dog function effect for change of test task threshold and priority.

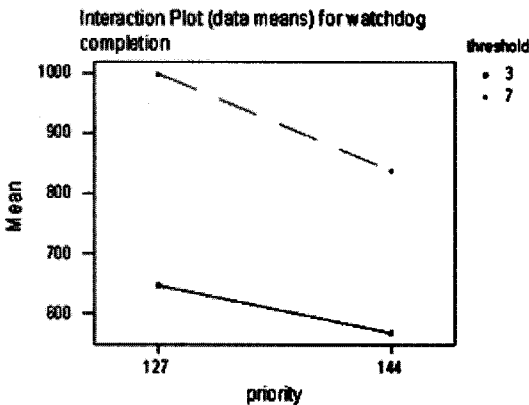


그림 8. Test Task의 임계 시간과 Priority의 상호 영향성.
 Fig. 8. Correlation of test task threshold and priority.

능의 정확성이 99.9 % 되며, 이는 시험치 결과와 일치함을 확인 할 수 있다.

Test Task의 임계 시간과 우선순위의 상호 영향성은 그림 7과 그림 8에서 보는 바와 같다. 즉 Test Task의

Priority 값보다 임계 시간 값이 Watch-Dog 기능에 주 영향을 미치게 되고, Test Task의 임계 시간과 Priority는 상호 교호 작용이 없음을 확인 할 수 있다.

5. 제안된 S/W Watch-Dog 기법의 정확성

본 논문에서는 기존의 Watch-Dog 기법과 제안된 Watch-Dog 기법이 시스템 부하에 미치는 영향을 평가하기 위하여 그림 1과 같은 환경을 구축하였다. AX-4000 시험 장비에서 10개의 E1 링크를 통해 Data Rate를 10Mbps~20Mbps까지 변화시키면서 기존의 S/W Watch-Dog 기법과 제안된 Watch-Dog 기법의 정확성을 측정하였다. 즉 Data Rate를 10Mbps~20Mbps 까지 변화시키면서 Threshold = 7, Priority = 127 인 Test Task를 수행한 경우, Watch-Dog 기능의 정확성(Test Task 구동 후 7초 되는 시점에 Unit Restart 여부)를 확인하였다.

그림 9에서 보는 바와 같이 Data Rate가 증가함에 따라 기존의 S/W Watch-Dog 기법(Single CPU 구조)에서는 하나의 프로세서에서 데이터 처리와 Watch-Dog 기능의 동시 수행으로 인해 Watch-Dog 기능의 정확성이 급격하게 떨어지는 반면에, 제안된 기법(DUAL CPU 구조)에서는 Master Processor는 Data 처리 Processor로부터 수신한 Message의 Sequence 값의 변화에 대한 Guard-Time으로 인해, Data 처리 Processor의 부하에는 거의 영향을 받지 않으며, Processor 자신의 부하 및 장애 따라 Watch-Dog 기능의 정확성이 결정됨을 알 수 있다.

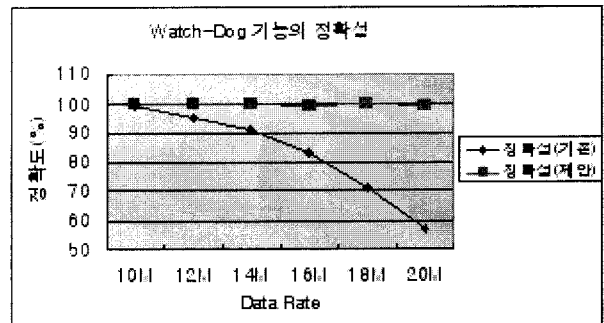


그림 9. 일반적인 Watch-Dog 기법과 제안된 기법에서의 Watch-Dog 기능의 정확성.
 Fig. 9. Accuracy of Proposed Watch-Dog method and general Watch-Dog method.

IV. 결 론

본 논문에서는 일반적인 S/W Watch-Dog 기법에 대해 살펴보고, 이를 바탕으로 상이한 OS로 구동되는 Dual CPU 기반에서 S/W Watch-Dog 기능의 신뢰성을 극대화 할 수 있는 방안을 제안하고, 제안된 기법에서 Watch-Dog 기능을 수행하는 Task의 임계 시간 및 우선 순위의 최적 값을 시험 및 분석 Tool을 사용하여 찾아내어, Data Rate의 변화에 따른 S/W Watch-Dog 기능의 정확성을 극대화 시킬 수 있었다.

본 논문에서 제안한 상이한 OS로 구동되는 Dual CPU 기반의 S/W Watch-Dog 기법으로 프로세서의 상태전단 및 복구 작업을 신속하게 처리할 수 있게 되어 데이터 서비스 망에 대한 신뢰성을 한층 더 높일 수 있을 것이다. 또한 향후 과제로 프로세서의 장애 발생에 대한 구체적인 정보를 저장하는 기능 및 장애 발생에 대한 사전 차단 기능에 대한 연구가 계속되어야 할 것이다.

참 고 문 헌

- [1] 고광호, 김봉수, 김한경, "SystemOAM 기능을 위한 개념 정립." 한국전자통신 연구소 전자통신동향 분석 제 8권 제 2호, 24-25쪽, 1998.7.
- [2] 삼성전자(주), 소용량 AceMAP 시스템 요구 기능서, 2001.1.
- [3] "draft-ietf-oam-requirements-01" 2003-04-18.

저 자 소 개



유 영 일(정회원)
 1999년 전북대학교 전자공학과
 학사 졸업.
 2001년 전북대학교 전자공학과
 석사 졸업.
 2001년 4월~현재 서울통신기술(주)
 통신시스템 연구소 연구원
 <주관심분야 : CDMA, W-CDMA, ETC, LTE>

전 병 실(정회원)
 대한전자공학회 논문지
 제39권(4월호) 논문2002-39TC-4-4 참조