DOI: 10.3745/KIPSTA.2008.15-A.6.309

# ASR 기법을 적용한 임무지향 교전통제 컴퓨터의 신뢰도 분석

신 진 범† · 김 상 하††


## 요 약

대공방어용 임무지향 교전통제 컴퓨터는 장시간 동안 임무의 중단없이 방어 임무를 수행하여야 하며, 복잡한 내장형 임무 소프트웨어를 탑재하는 컴퓨터에는 대공방어 임무의 특성상 확실성과 안정성 및 신뢰성을 보장하여야 한다. 구현된 임무지향 교전통제 컴퓨터에서 임무수행의 확실성과 안정성은 4 장의 프로세서로 구성되는 분산 컴퓨터 아키텍처에 의해 보장되며, 신뢰도는 분산 구조의 컴퓨터에 저비용의 능동 예비 이중화(ASR) 고장감내 기법을 적용하여 보장되도록 하였다. 소프트웨어적인 능동 예비 이중화 고장감내 기법은 높은 신뢰도와 신속한 고장복구 성능을 가지는 교전통제 컴퓨터를 저비용으로 구현하므로 대공방어용 컴퓨터에 매우 적합한 기법이다. 본 논문은 능동 예비 이중화 고장감내 기법의 메커니즘과 성능분석에 대해 기술하고, 교전통제 컴퓨터에 ASR 기법과 하드웨어적인 DMR 및 TMR 고장감내 기법을 적용한 경우의 MTBF, 신뢰도, 가용성 및 저비용성을 비교분석하였다. ASR 기법은 72 시간의 임무 시간에 대하여 TMR과 유사한 임무 신뢰도를 제공하며, 저비용의 구현이 가능하므로 교전통제 임무지향 컴퓨터의 고장감내 기법으로 최적인 것으로 분석되었다.


**키워드 : 능동 예비 이중화, 고장감내, 고장탐지, 고장복구, 임무지향 교전통제 컴퓨터, 대공방어, 임무 신뢰도**

# Reliability Analysis of The Mission-Critical Engagement Control Computer Using Active Sparing Redundancy

Shin, Jin Beom† · Kim, Sang Ha††


## ABSTRACT

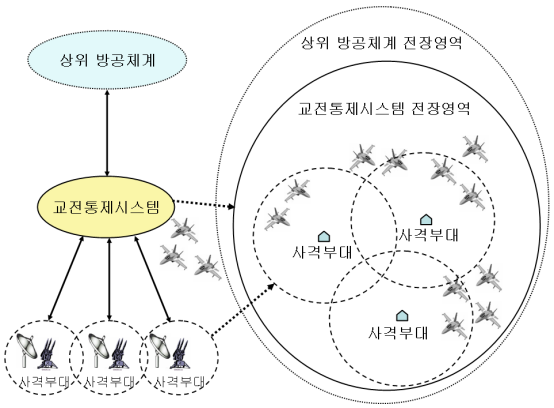
The mission-critical engagement control computer for air defense has to maintain its operation without any fault for a long mission time. The mission performed by large-scale and complex embedded software is extremely critical in terms of dependability and safety of computer system, and it is very important that engagement control computer has high reliability. The engagement control computer was implemented using four processors. The distributed computer composed of four processors quarantees the dependability and safety, and ASR fault-tolerant technique applied to each processor guarantees the reliability. In this paper, the mechanism and performance of ASR fault-tolerant technique are analysed. And MTBF, reliability, availability, and cost-effectiveness for ASR, DMR and TMR techniques applied to the engagement control computer are analysed. The mission-critical engagement control computer using software-based ASR fault-tolerant technique provides high reliability and fast recovery time at a low cost. The mission reliability of the engagement control computer using ASR technique in 4 processors board is almost same the reliability of the computer using TMR technique in 6 processors board. ASR technique is most suitable to the mission-critical engagement control computer.


**Keywords : Active Sparing Redundancy, Fault Tolerant, Fault Detection, Fault Recovery, Mission-Critical Engagement Control Computer, Air Defense, Mission Reliability**

## 1. 서 론

대공방어용 교전통제 시스템은 (그림 1)의 전장에서 아군 및 주요시설을 공격하는 다수의 적기에 대해 사격부대에 사격을 명령하고, 통제하는 교전임무를 수행한다. 교전임무는 교전통제 컴퓨터에서 장시간에 걸쳐 실시간적으로 수행되므로, 컴퓨터의 고장은 대공방어 임무의 마비를 초래하여 아군의 심각한 피해를 초래하게 된다. 이런 임무지향(Mission-Critical) 컴퓨터에는 고장감내 기능이 필수적[1,2,3]이며, 최근 하드웨어 기술의 발달로 컴퓨터 시스템의 고장율은 $10^{-5}$ 정도에서 $10^{-10}$ 정도로 감소하였지만 복잡한 대규모의 임무 소프트웨어를 내장하는 컴퓨터에는 안정성과

† 정 회 원 : 국방과학연구소 책임연구원
†† 종신회원 : 충남대학교 전기정보통신공학부 교수
논문접수 : 2008년 6월 3일
수 정 일 : 1차 2008년 9월 10일
심사완료 : 2008년 9월 18일




(그림 1) 교전통제 시스템의 전장

신뢰성이 필수적으로 요구된다[4,5].

고장감내 기법은 컴퓨터의 임무시간과 운용환경[6]에 따라 적합한 방법이 적용되어야 한다. 지상 운용 장비인 교전통제 시스템은 전투기 보다는 길고 우주항공 장비보다는 짧은 수 일 정도의 임무지속 시간과 실시간의 고장복구 성능이 요구되므로 전투기의 컴퓨터와 같은 임무지향 고장감내 특성이 요구된다. 또한 자동복구가 불가한 치명적인 고장은 임무를 중단하고, 2 시간의 MTTR(Mean Time To Repair) 내에 수리부속으로 정비가 가능하므로 지상장비에 적합한 저비용의 고장감내 기능 구현이 요구된다.

교전통제 컴퓨터의 고장감내 아키텍처에는 ASR(Active Sparing Redundancy) 고장감내 기법과 네트워크 이중화 기법이 적용되었다. ASR 기법은 소프트웨어 고장감내 기법으로 주 소프트웨어와 동시에 동작하는 예비 소프트웨어에 의해 고장복구를 수행하며, 기본 아키텍처에 비해 하드웨어의 추가없이 저비용으로 고장감내 기능의 구현이 가능하다. 네트워크 이중화는 통신량이 많은 교전통제 컴퓨터에 안정성과 신뢰성을 제공한다. ASR 기법과 네트워크 이중화를 적용한 교전통제 컴퓨터는 임무시간을 만족하는 신뢰도를 제공하고, 지상장비에 적합한 저비용의 고장감내 기능을 제공한다.

본 논문에서는 ASR 기법과 네트워크 이중화를 적용한 교전통제 컴퓨터의 고장감내 아키텍처 설계와 구현, 그리고 ASR 고장감내 기법의 메커니즘을 기술하였으며, 고장감내 성능과 MTBF(Mean Time Between Failure) 및 임무신뢰도(Mission Reliability)를 분석하였다. 또한 ASR 기법과 네트워크 이중화가 적용된 교전통제 컴퓨터의 신뢰도와 DMR(Dual Modular Redu-ndancy) 기법과 TMR(Triple Modular Redundancy) 기법을 적용한 컴퓨터의 신뢰도를 비교 분석하였고, ASR 기법의 저비용 구현성을 입증하였다.

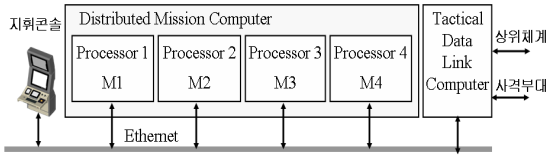
## 2. 고장감내 아키텍처 설계

### 2.1 교전통제 컴퓨터의 기본 아키텍처

수백 개의 트랙을 실시간으로 처리하면서 많은 정보를 교환하는 교전임무는 교전통제 컴퓨터에 내장되는 대규모의 복잡한 임무 소프트웨어[7]에 의해 수행된다. 복잡한 내장형(Embedded) 소프트웨어의 안정성을 위하여 임무 소프트웨어는 <표 1>과 같이 4 장의 프로세서에 M1, M2, M3 및 M4로 분산되며, 기본 컴퓨터의 아키텍처는 (그림 2) 와 같다. 분산 구조는 임무의 처리 부하와 통신 대역폭을 분산하여 안정성을 제공한다. 프로세서에서 출력되는 정보들은 브로드캐스트 방식으로 모든 프로세서로 전달되며, 이더넷 네트워크의 전체 통신 대역폭은 약 35.2 Mbyte 이다. 작전 임무 수행시에 교전통제 컴퓨터는 전술데이터 링크(Tactical Data Link) 컴퓨터와 연결되어 동작하며, 훈련시에는 M4 프로소프트웨어가 전술데이터링크 컴퓨터를 모의한다. 그러므로 M4의 통신 대역폭은 전술데이터링크 컴퓨터의 대역폭과 동일하다.

〈표 1〉 교전통제 소프트웨어의 분산

| 구 분 / 소프트웨어 | 프로세서 구분 | 기능 | 통신대역폭 (Mbyte) |
|---|---|---|---|
| M1 | 1 | 전장관리 | 0.8 |
| M2 | 2 | 트랙관리 | 1.5 |
| M3 | 3 | 위협평가/무기할당 교전통제 | 2.5 |
| M4 | 4 | 상위방공체계 및 사격부대 모의 | 30.4 |



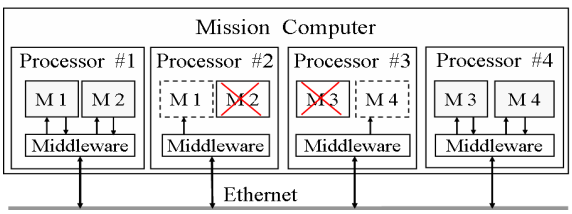

(그림 2) 교전통제 컴퓨터의 기본 아키텍처

고장율이 λ인 프로세서로 구성되는 기본 컴퓨터의 MTBF는 식 (1)과 같다.

$$MTBF_{BASIC} = \frac{1}{4\lambda} \quad (1)$$

### 2.2 ASR 고장감내 아키텍처

ASR 기법은 2-Version 고장감내[8] 기법과 같은 소프트웨어 고장감내 기법으로 이를 적용한 컴퓨터 아키텍처는 (그림 3) 과 같다. ASR 기법은 프로세서 #1의 주 소프트웨




(그림 3) ASR 기법의 고장감내 아키텍처

어 모듈 M1과 동일한 이중화 예비 모듈 M1 을 프로세서 #2 에 내장하고 두 모듈의 동작상태를 감시하여 고장을 복구한다. 고장시 주 모듈은 분리되고 예비 모듈이 연결되어 동작하며, ASR 기법은 (그림 2)의 기본 아키텍처와 동일한 수량의 프로세서로 구현되므로 저비용의 구현이 가능하다.

### 2.3 네트워크 이중화

교전통제 컴퓨터의 프로세서에 내장되는 임무 소프트웨어들은 이더넷에 기반하여 많은 정보를 상호 교환하므로 네트워크의 안정성과 신뢰성은 중요하다. 그러므로 이더넷 네트워크를 이중화로 구성하여 ASR 기법과 함께 고장감내 기능을 제공하였다.

### 2.4 기본 가정

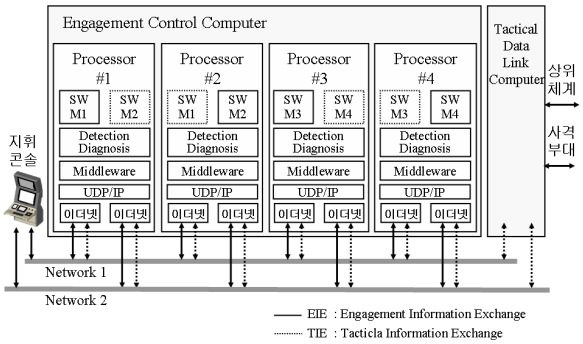
소프트웨어 고장감내 방법인 ASR 기법을 적용하여 교전통제 컴퓨터의 고장감내 기능을 구현하기 위해서는 다음의 가정이 필요하다. 이는 컴퓨터의 고장이 내부의 프로세서에 의해 야기된 것으로 제한한다.

- 소프트웨어 코드에는 오류가 없다. 따라서 소프트웨어에서 탐지되는 고장이란 하드웨어의 고장이다.
- 외부 네트워크와 외부 장비에는 오류가 없다. 정보 전달 상의 오류는 프로세서 하드웨어의 고장이다.
- 컴퓨터 내부의 케이블과 모체기판 및 전원 모듈에는 고장이 없다.

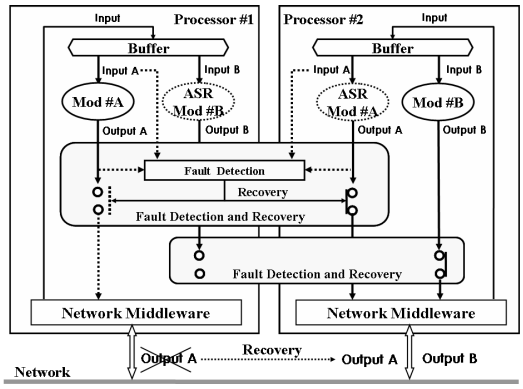
### 2.5 교전통제 컴퓨터의 고장감내 아키텍처

ASR 기법과 네트워크 이중화를 적용하여 설계된 교전통제 컴퓨터의 아키텍처는 (그림 4)와 같다. 각 프로세서의 임무 소프트웨어와 물리 네트워크 사이에는 소프트웨어들의 안정적인 동작을 위하여 QoS(Quality Of Service)와 리소스 관리를 제공[9,10]하는 네트워크 미들웨어가 탑재되며, 네트워크 미들웨어 위에는 고장탐지와 진단 및 복구를 위한 소프트웨어가 탑재된다.

소프트웨어들의 교환 정보들은 프로세서들과 지휘콘솔의 교환정보인 교전정보(EIE)와 프로세서들과 전술데이터링크 컴퓨터의 교환정보인 전술정보(TIE)로 구분된다. 정보에는 오류검사 코드(Checksum Code)가 포함되어 있으므로 정보




(그림 4) 교전통제 컴퓨터의 고장감내 아키텍처




(그림 5) ASR Fault Tolerant Model

의 수신시에 오류의 점검이 가능하다. 전술정보는 전술데이터 링크 컴퓨터를 통하여 외부 시스템과 교환된다.

## 3. ASR 고장감내 메커니즘

### 3.1 ASR 고장감내 모델

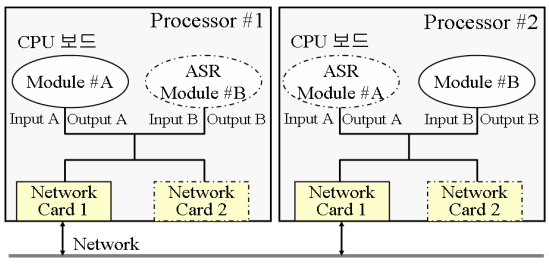
두 장의 프로세서에 내장된 소프트웨어 주 모듈과 ASR 예비 모듈은 (그림 5)와 같이 외부에서 입력된 입력 A를 동시에 수신한다. 입력 A에는 오류검사 코드가 포함되어 있으므로 오류의 점검이 가능하다. 두 모듈은 입력을 처리한 후 오류 검사 코드를 포함한 결과로 출력 A를 출력한다. 고장탐지 모듈은 주 모듈과 이중화 모듈 #A의 입력 입력 A를 상호 비교하고, 출력 출력 A를 상호 비교하여 고장을 탐지한다. 고장탐지 결과와 CPU 계통의 고장을 주기적으로 점검하는 Heartbeat 결과에 의해 CPU 계통과 네트워크 카드 계통의 하드웨어 고장을 진단한다. 고장진단 결과는 CPU 계통이나 네트워크 카드 계통의 고장을 다른 프로세서에서 동작하는 이중화 모듈로 복구하거나 이중화네트워크 카드로 복구한다. 이 절차는 컴퓨터를 구성하는 모든 쌍의 프로세서에 대해 수행된다.

### 3.2 동기화

각기 다른 프로세서에서 동작하는 주 소프트웨어 모듈과 능동 이중화 모듈은 거의 동일한 시간에 입력정보를 수신하며, 입력 처리후 거의 동일한 시간에 출력정보를 출력하게 된다. 고장탐지는 두 모듈의 출력이 모두 수신되었거나, 출력의 수신 Timeout이 발생한 경우에 입출력 정보를 비교하여 수행되므로 이 시점에 두 모듈의 동기화가 이루어진다.

### 3.3 고장탐지 및 고장진단

주 모듈과 이중화 모듈이 내장된 두 장의 프로세서에 대해 외부의 입력 정보는 (그림 6) 과 같이 네트워크 카드를 통하여 CPU 보드로 수신되고, 처리 결과는 CPU 보드에서 네트워크 카드를 통하여 외부로 출력된다. 프로세서의 고장은 CPU 보드의 고장과 네트워크 카드의 고장에 기인하므로, 두 프로세서에서 동작하는 모듈 #A와 이중화 모듈 #A의

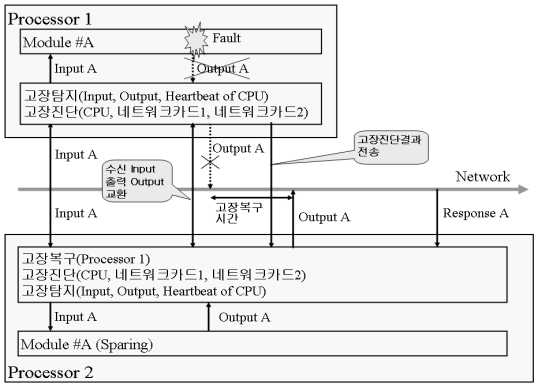



(그림 6) 프로세서 동작 모델

입력과 출력 들을 상호 비교하고, 두 장의 CPU 보드들의 Heartbeat 결과를 비교하여 CPU 보드에 고장이 있는지 네트워크 카드에 고장이 있는 지를 진단할 수 있다. 네트워크 카드 1의 고장은 네트워크 카드 2로 복구되며, 네트워크 카드 2의 고장이나 CPU 보드의 고장은 이중화 모듈이 동작하는 프로세서에 의해 복구된다. Heartbeat 는 CPU와 메모리와 관련된 정보 입출력 기능들이 정상적으로 동작하는 지를 100 msec의 주기로 기능적으로 점검하여 임무 수행 중에 고장이 있는지를 탐지하며, 메모리와 입출력 장치를 포함한 CPU의 모든 기능은 점검하지 않는다. 모든 메모리의 점검을 포함한 CPU의 세부적인 점검은 교전통제 컴퓨터에 전원이 인가되는 초기에 점검되며, 이런 고장은 초기에 제거된다.

### 3.4 고장복구

고장진단은 프로세서의 고장이 네트워크 카드에 있는지 CPU 보드에 있는지를 결정하며, 주 프로세서의 고장진단 결과는 이중화 모듈 탑재 프로세서로 전달된다. 주 프로세서의 네트워크 카드 1의 고장은 네트워크 카드 2에 의해 복구되며, 네트워크 카드 2의 고장과 CPU 보드의 고장은 이중화 모듈이 탑재된 프로세서에 의해 복구된다. (그림 7)은 입력의 수신 후 발생한 주 모듈의 고장을 예비 모듈이 복구하여 출력을 수행하는 과정을 보여준다. 이중화 모듈 탑재 프로세서는 주 프로세서의 고장진단 결과를 전송받아(수신 없을 시는 Timeout 처리) 주 프로세서에 고장(네트워크 카드 2 또는 CPU 보드 고장)이 있다는 것을 알게되어 이중화 모듈의 출력이 네트워크로 전송되도록 한다. 주 프로세서가 CPU 보드의 Heartbeat 결과와 함께 이중화 모듈 탑재 프로세서에서 보내온 입출력 정보를 비교하여 고장이 없다고 판단하였을 경우, 주 프로세서는 출력 정보를 네트워크로 전송한다. 그러므로 이 시점부터 이중화 모듈 탑재 프로세서의 고장복구 모듈이 출력 정보를 네트워크로 출력하는 시간까지의 고장복구 시간이 소요된다. 예비 모듈은 주 모듈과 동기를 지속적으로 유지하므로 복구 시점의 결정을 위한 별도의 절차는 필요하지 않다.




(그림 7) ASR 모듈에 의한 고장복구

## 4. 교전통제 컴퓨터의 구현

교전통제 컴퓨터는 (그림 4)와 같이 4장의 프로세서로 제작되었으며, 프로세서의 CPU 보드는 VME-183(Pentium 1.2GHz, Curtiss Wright)이고 이더넷 네트워크 카드는 GNET/PMC(1Gbyte, CCII)이다. 임무 소프트웨어들과 고장탐지, 고장진단 및 고장복구 소프트웨어 모듈이 구현되어 각 프로세서에 탑재되었다.

교전통제 컴퓨터의 고장감내 성능은 임무 소프트웨어와 CPU 보드의 Heartbeat 검사 소프트웨어에 랜덤한 고장을 발생하는 모의 프로그램을 추가하여 시험되었다. 임무 소프트웨어에 추가된 고장 모의 프로그램에는 입력 정보의 수신 부분과 처리 결과의 출력 부분에서 고장이 발생하도록 하여 CPU 보드와 네트워크 카드의 고장을 모의하였다. 10,000 회 이상의 모의 고장에 의한 시험에서 교전통제 컴퓨터는 임무 처리 주기인 1,500 msec 의 시간 내에 고장을 복구하였다.

## 5. 분 석

### 5.1 고장탐지, 고장진단 및 고장복구 분석

ASR 기법이 구현된 두 장의 프로세서에서 CPU 보드와 네트워크 카드의 고장탐지는 주 소프트웨어 모듈과 이중화 모듈의 입출력의 동일성과 Heartbeat 결과 및 입력의 오류검사를 비교하여 수행된다.

(그림 6)에서 프로세서 #1의 CPU 보드의 고장은 <표 2>와 같이 프로세서 #1과 #2의 모듈 #A의 출력 를 서로 다르게 하고, 모듈 #B의 출력를 서로 다르게 한다. 또한 프로세서 #1과 #2의 모듈 #A로 입력되는 입력 A를 다르게 하고 프로세서 #1과 #2의 모듈 #B로 입력되는 입력 B를 서로 다

〈표 2〉 고장탐지 및 고장진단 표

| 진단내용 / 고장유형 | 프로세서 1과 2의 출력 동일성 | 프로세서 1과 2의 입력 동일성 | CPU 계통 Heartbeat Check | 입력 Checksum Error | 예비 네트워크 카드 |
|---|---|---|---|---|---|
| 프로세서 CPU 보드 Fault | × | × | ○ | × or ○ | × or ○ |
| 네트워크 카드 2 Fault | × | × | × | ○ | × |
| 네트워크 카드1 Fault (네트워크 카드2 정상) | × | × | × | ○ | ○ |

르게 하며, CPU 보드의 Heartbeat 오류를 발생시킨다. 그러므로 이 탐지결과들을 비교하면 프로세서 #1의 CPU 보드에 고장이 있는 것으로 진단된다.

프로세서 #1의 네트워크 카드 1의 고장은 프로세서 #1과 #2의 모듈 #A의 입력을 다르게 하고 출력 을 서로 다르게 하며, 프로세서 #1과 #2의 모듈 #B에 대해서도 입력을 다르게 하고 출력을 다르게 한다. 이 경우 프로세서 #1의 입력 A와 입력 B에 오류검사 코드에 의해 고장이 탐지되지만, CPU 보드의 Heartb-eat에는 오류가 발생하지 않는다. 따라서 네트워크 카드 1의 고장으로 진단된다. 이때 예비 네트워크 카드2 의 유무도 진단에 포함된다.

주 소프트웨어 모듈이 내장된 프로세서에 탑재된 고장진단 소프트웨어 모듈은 주 모듈과 ASR 예비 모듈의 고장탐지 결과와 Heartbeat 결과를 비교하여 고장진단 결과로 네트워크 카드 또는 CPU 보드의 고장을 알려준다. 네트워크 1 카드의 고장은 네트워크 카드 2에 의해 복구되며, 네트워크 카드 2 또는 CPU 보드의 고장은 프로세서 2의 이중화 모듈로 고장을 복구한다. CPU 보드나 네트워크 카드가 전혀 동작하지 않는 하드웨어의 치명적인 고장에 대한 진단은 고장진단 소프트웨어의 Timeout에 의해 처리된다.

### 5.2 고장복구 시간 분석

교전통제 컴퓨터의 고장감내 성능은 임무 소프트웨어와 Heartbeat 소프트웨어에서 랜덤하게 생성된 모의 오류에 의해 시험되었다. 모의 오류는 임무 소프트웨어에서 정보의 입력 부분과 출력 부분에 오류 프로그램을 추가하고 Heartbeat 소프트웨어에 오류 프로그램을 추가하여 생성되도록 하였다. 그리고 모의 오류 프로그램은 CPU 보드나 네트워크 카드가 전혀 동작하지 않는 치명적인 고장도 모의하도록 하였다.

랜덤하게 발생시킨 10,000 회 이상의 모의 고장에 대해 고장탐지와 고장진단을 포함하여 고장이 복구되기 까지의 시간은 1,500 msec 이하로 측정되었으며, 교전통제 컴퓨터에 요구되는 실시간적 고장복구 성능 조건을 만족하였다.

### 5.3 MTBF 및 가용도 분석

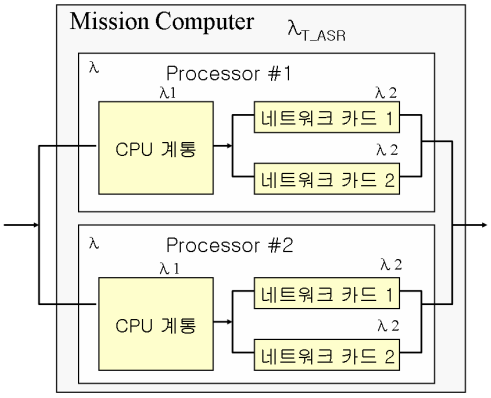
ASR 기법이 적용된 2장의 프로세서에 대한 신뢰도 모델은 (그림 8)과 같으며, 고장율이 λ인 모듈의 능동 이중화에 의한 MTBF는 3/(2λ) 이므로 고장율 λTASR에 대한 MTBFTASR 은 식 (2)와 같이 표현된다[11]. m은 프로세서의 고장율 λ 에 대한 CPU 보드의 고장비율이고 n 은 네트워크 카드의 고장비율이다. 교전통제 컴퓨터에 사용하는 CPU 보드[12] 와 네트워크 카드[13]의 고장율 값으로부터 m은 0.9, n은 0.1 정도로 나타낼 수 있다. 식 (2)로부터 4장의 프로세서로 구성되는 교전통제 컴퓨터의 MTBFASR 는 식 (3)과 같이 계산된다.

$$\lambda_1 = m\lambda, \quad \lambda_2 = n\lambda$$

$$\lambda_{T\,ASR} = \frac{2}{3}(m + \frac{2}{3}n)\lambda$$

$$MTBF_{TASR} = \frac{3}{2(m + \frac{2}{3}n)\lambda} \tag{2}$$

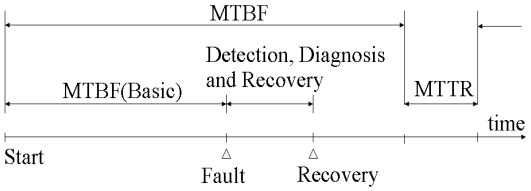
$$MTBF_{ASR} = \frac{90}{116\ \lambda} \tag{3}$$




(그림 8) ASR 아키텍처의 신뢰도 모델

ASR 기법을 적용한 컴퓨터의 MTBF는 기본 컴퓨터의 MTBF 보다 3 배 정도 증가한다. MTTR은 2 시간 이하로 할당되었으므로, (그림 9)와 같이 기본 컴퓨터의 MTBF(Basic)에 비해 MTBF가 증가하고, 이에 따라 가용도(Availability)[14]도 증가한다. 고장이 발생후에 고장복구까지의 시간은 1,500 msec 이하이므로 MTBF 나 가용도에 영향을 거의 미치지 않는다.

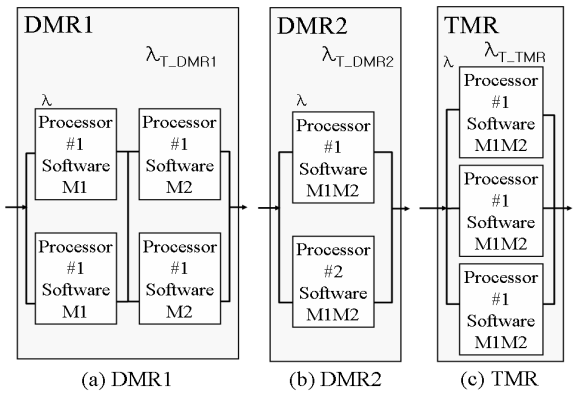
$$\text{가용도} = \frac{MTBF}{MTBF + MTTR}$$




(그림 9) 교전통제 컴퓨터의 MTBF 개선

### 5.4 DMR 및 TMR 기법과의 MTBF 비교 분석

교전통제 컴퓨터를 구성하는 2장의 프로세서에 대해 네트워크의 이중화 없이 DMR 및 TMR 기법을 적용한 신뢰도 모델은 (그림 10)과 같다. (그림 10)의 (a)는 (그림 2)의 기본 아키텍처에 DMR 기법을 적용한 구조이며, (b)와 (c)는 두 개의 주 소프트웨어 모듈 M1과 M2를 내장한 프로세서에 대해 DMR 및 TMR 기법을 적용한 구조이다. 교전통제 컴퓨터를 구상하는 4장의 프로세서에 이들기법을 적용하였




(그림 10) DMR 및 TMR 신뢰도 모델

〈표 3〉 MTBF 및 가용도 비교표

| 구분 / 아키텍처 | MTBF(Hr) | Availability |
|---|---|---|
| ASR | 37,202 | 0.99995 |
| 기본 컴퓨터 | 11,673 | 0.99983 |
| DMR1 | 17,509 | 0.99989 |
| DMR2 | 35,018 | 0.99994 |
| TMR | 42,800 | 0.99995 |

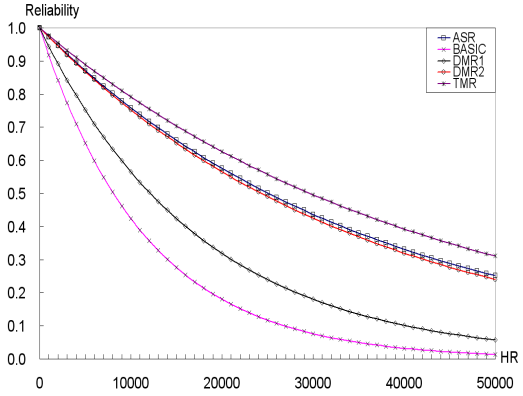
을 경우의 MTBF는 식 (4), (5) 및 (6) 과 같다. 그림 (b)의 DMR1 기법의 경우 2장의 프로세서에 대한 MTBF는 앞서 5.3에 기술한 바와 같이 3/(2λ) 이므로 4장에 대해서는 3/(4λ) 가 된다[11].

$$MTBF_{DMR1} = \frac{3}{8\lambda} \quad (4)$$

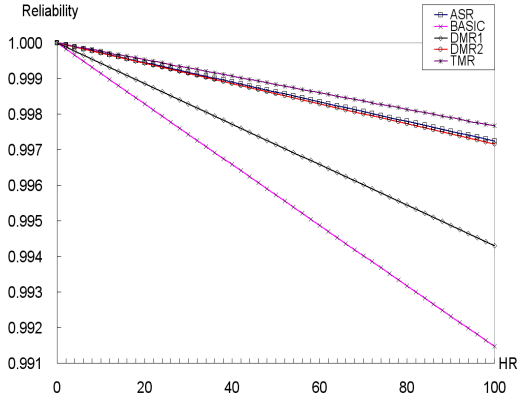
$$MTBF_{DMR2} = \frac{3}{4\lambda} \quad (5)$$

$$MTBF_{TMR} = \frac{11}{12\lambda} \quad (6)$$

프로세서의 고장율(λ)은 제조회사에서 제공되는 CPU 보드[12]와 네트워크 카드[13]의 고장율로부터 네트워크 이중화가 적용된 프로세서의 경우 $2.08556 \times 10^{-5}$ 으로, 네트워크 이중화가 없는 프로세서의 경우 $2.14175 \times 10^{-5}$ 으로 계산되므로 각 기법을 적용한 교전통제 컴퓨터의 MTBF는 식 (1), (3), (4), (5) 및 (6)으로부터 <표 3>과 같이 계산되며, 가용도는 5.3에 따라 계산된다. ASR 기법의 MTBF는 DMR 기법들 보다는 크지만 TMR 기법 보다는 작다. ASR 기법의 MTBF 가 DMR2 보다 큰 이유는 네트워크가 이중화되었기 때문이며, 이 이중화를 제외하면 DMR2와 동일하다. 기본 아키텍처와 DMR1 기법은 신뢰도는 작지만 리소스 예비율을 보장하는 장점이 있으며, DMR2와 TMR 기법은 리




(그림 11) 교전통제 컴퓨터의 임무 신뢰도




(그림 12) 교전통제 컴퓨터의 100 시간 신뢰도

소스 예비율을 보장하지 못한다. ASR 기법의 경우는 주 모듈과 이중화 모듈중 하나만이 외부와 통신을 수행하므로 통신 대역폭에 대한 리소스를 보장하는 장점이 있다.

### 5.5 임무 신뢰도 분석

교전통제 컴퓨터의 임무 신뢰도는 다음과 같이 표현된다.

$$R_{(t)} = e^{-\frac{1}{MTBF}t}$$

기본 아키텍처, ASR, DMR1, DMR2 및 TMR 기법을 적용한 컴퓨터의 임무 신뢰도는 (그림 11)과 같으며, 72 시간의 임무시간에 대한 세부적인 신뢰도는 (그림 12)와 같다.

### 5.6 비용 분석

ASR 기법을 적용하여 4장의 프로세서로 구성되는 컴퓨터의 MTBF는 본 논문에서 기술한 6장의 프로세서를 사용하는 TMR 기법의 컴퓨터에 비해 5,598 시간이 적지만 교전통제 컴퓨터의 임무시간인 72 시간의 시점에서 신뢰도는 약 99.81 %이므로 TMR 기법의 신뢰도인 99.83 %와 거의 동일하다. 교전통제 컴퓨터는 임무시작 전의 예방정비 및 전원인가 전의 점검을 통해 정상적인 동작을 보장하며, 자동복구가 불가한 치명적인 고장은 2시간의 정비를 통해 임무

의 재수행이 가능하다. 그러므로 4장의 프로세서를 사용하는 ASR 기법은 6장의 프로세서를 사용하여 0.02 %의 임무 신뢰도가 개선되는 TMR 기법에 비하여 비용 측면에서 교전통제 컴퓨터의 고장감내 아키텍처로 매우 적합하다. 프로세서를 구성하는 CPU 보드 및 네트워크 카드의 가격은 3천만원 정도이며, 배치되는 컴퓨터의 수량에 따라 비용은 증가하게 된다.

## 6. 결 론

대공방어를 위해 다수의 위협 적기와 교전 임무를 수행하는 교전통제 컴퓨터의 고장은 적의 공격으로부터 아군의 큰 피해를 초래하므로 컴퓨터에는 고장감내 기능이 요구되었으며, 저비용의 고장감내 기능 구현이 가능한 ASR 고장감내 기법과 네트워크 이중화를 적용하여 교전통제 컴퓨터의 고장감내 기능을 구현하였다.

본 논문에서는 ASR 고장감내 기법의 메커니즘을 기술하였고, 적용된 기법의 고장탐지와 고장진단 및 고장복구 성능을 분석하였으며, MTBF 및 임무 신뢰도를 분석하였다. 그리고 교전통제 컴퓨터에 하드웨어적인 DMR 및 TMR 고장감내 기법들을 적용한 경우의 MTBF와 신뢰도를 ASR 기법의 값들과 비교 분석하여 ASR 기법의 저비용 구현성에 대하여 입증하였다.

ASR 기법은 복잡한 내장형 소프트웨어로 구성되는 지상장비용 컴퓨터 시스템에 매우 적합한 고장감내 기법이므로 향후 유사한 컴퓨터 시스템에 적용될 수 있을 것으로 사료된다. 향후에 소프트웨어의 처리 부하량과 네트워크 대역폭에 대한 리소스 예비율과 부하에 따른 안정성에 대한 추가적인 연구가 필요하다고 판단된다.

## 참 고 문 헌


[1] John Rushby, “Bus Architectures For Safety- Critical Embedded Systems,” Lecture Notes in Computer Science, Vol.2211, pp.306-323, October, 2001.

[2] ‘System Safety Management Guide’, Departm-ent of the Army, Pamphlet 385-16, 1987.

[3] ‘Risk Management’, Department of the Army, Field Manual No.100-14, April, 1998.

[4] Gul N. Khan, “Fault-Tolerant Architectures for High Performance Embedded System Applicat-ions”, in Proc. of IEEE Conference on Comp-uter Design, pp.384-389, October, 1998.

[5] K. H. (Kane) Kim, “Issues Insufficiently Reso-lved in Century 20 in the Fault Tolerance Distributed Computing Field”, in Proc. of the IEEE CS 19th Symp. Reliable Distributed Systems (SRDS), pp.106-115, October, 2000.

[6] Dhiraj K. Pradhan, ‘Fault-Tolerant Computer System Design’, p.6-7, Prentice-Hall, Inc., New Jersey, 1996.

[7] 유명환, 배정일, 신진화, 조길석, “UML 2.0 모델 기반의 교전통제 소프트웨어 아키텍처 개발” 한국군사과학기술학회지, 제10권, 제4호, pp.20-29, 12, 2007.

[8] Liming CHEN, Algirdas AVIZIENIS, “N-Vers-ion Programming : A Fault-Tolerant Approach To Reliability Of Software Operation,” in Proc. of the IEEE Fault Tolerant Computing Systems, Vol.3, pp113-119, 1996.

[9] K. H.(Kane) Kim, “Toward Globally Optimal Resource Management in Large-Scale Real Time Distributed Computer Systems,” in Proc. of the IEEE CS 6th Workshop on Future Trends of Distributed Computing Systems, pp.248-255, October, 1997.

[10] K. H.(Kane) Kim, “Middleware of Real-Time Object Based Fault Tolerance Distributed Computing Systems: Issues And Some Appr-oaches”, in Proc. of the Pacific Rim Int’l Symp. on De-pendable Computing, pp.3-8, December, 2001.

[11] Norman B. Fuqua, ‘Reliability Engineering for Electronic Design’, pp.135-137, MARCEL DEKKER, INC., New York, 1987.

[12] http://www.cwcembedded.com/p/5/3/75.html

[13] http://www.ccii.co.za/products/blp_pmc/pmc_fc-1gbps.html

[14] Dhiraj K. Pradhan, ‘Fault-Tolerant Computer System Design’, pp.70-73, Prentice-Hall, Inc., New Jersey, 1996.



신 진 범

e-mail : espinosa@hanafos.com

1987년 부산대학교 전자공학(석사)

1987년～현 재 국방과학연구소 책임연구원

관심분야 : 실시간시스템, 컴퓨터구조, 임베디드시스템 및 네트워크


## 김 상 하

e-mail : shkim@cnu.ac.kr

1980년 서울대학교(석사)

1986년 미국 휴스턴대학교(석사)

1989년 미국 휴스턴대학교(박사)

1990년~1992년 한국과학기술연구원 선임연구원

1990년~현 재 충남대학교 전기정보통신공학부 교수

관심분야 : 컴퓨터구조, 임베디드 시스템, 정보통신, 컴퓨터네트워크