
VMEbus를 통한 이중화 네트워크 프로토콜 구현

박정원* · 박성진**

Implementation of a redundant network protocol based on VMEbus

Jeong-Weon Park* · Seong-Jin Park**

요 약

군의 요구에 의해서 장비 성능에 대한 안정성과 긴박한 시간에 그 성능을 유지할 수 있는 생존성을 증대시키기 위한 방법이 대두되고 있으며, 그 방법 중의 하나로 시스템에서의 이중화 설계에 대한 이슈가 늘어나고 있는 추세이다. 일반적으로 시스템의 생존성을 증대시키기 위한 방법으로써 적용하는 이중화 기법은 두 개의 프로세스 상호간에 두 개의 네트워크망을 구성하여 이중화를 구현하는 것이다. 그러나 프로세스의 고장이나 물리적 네트워크망이 손실되었을 경우 기능을 제대로 수행하지 못할 수 있다. 이에 본 논문에서는 VMEbus의 master와 slave 간의 공유 메모리 영역, 인터럽트 방식 적용, 이중화를 담당하는 전용 task와 통신 이상 시 이를 처리하는 이벤트를 발생시키는 프로토콜을 직접 구현하고, 실험을 통하여 이 방안의 타당성을 확인한다.

ABSTRACT

According to the needs for equipment performance, reliability, and performance for impending time to guarantee its viability, various redundancy techniques have been used in many military systems. Typically, one of redundancy schemes to increase the viability of the system is to configure a network between two processes. However, when some failure or loss in a physical network occur, redundancy schemes may not operate well. In this paper, we present a protocol scheme of VMEbus master and slave of a shared memory region, interrupts, and a dedicated task communication. Specifically, we confirm the validity of the presented scheme through the direct implementation of the protocol and some experimental results.

키워드

이중화, 네트워크 프로토콜, VMEbus

Key word

Redundant, Network Protocol, VMEbus

* 정회원 : LIG넥스원(주) (주저자, ephome@naver.com)
** 정회원 : 아주대학교 전자공학과 교수 (교신저자)

접수일자 : 2010. 06. 04
심사완료일자 : 2010. 08. 07

I. 서 론

국내에서 개발하여 운용중인 군용 장비에 적용한 이중화의 기법은 장비 내부에서 주 업무를 수행하는 프로세서를 이중화하여 적용하는 방식이다. 이 방법은 하나의 네트워크망이 손실되었을 경우 다른 네트워크망을 통하여 통신을 지속적으로 수행한다. 그러나 프로세서 자체가 비정상 또는 기능을 상실하게 되면 더 이상의 업무를 수행할 수가 없게 된다. 이러한 경우 군용 장비에서는 치명적인 운용 손실을 가져올 수 있다.

본 논문에서는 이를 보완하기 위하여 우선적으로 두 개의 프로세서를 적용한다. 그러나 두 번째 프로세서가 백업 기능을 수행하지는 않는다. 그리고 각각의 프로세서가 갖고 있는 두 개의 네트워크망에서 하나씩 외부 장비와 우선적으로 통신을 수행한다. 여기서 두 개의 프로세서 상호 간에 동일하게 네트워크망을 적용하지 않고, VMEbus 통신을 통한 이중화 네트워크 프로토콜망을 구성한다. 이는 VMEbus에서의 Master 프로세서와 Slave 프로세서 간의 공유 메모리 영역의 사용과 이를 위한 인터럽트 방식을 적용하고, 이중화를 담당하는 전용 Task와 통신 이상 시 이를 처리하는 이벤트를 구현한다. 이로써 내부에서는 두 개의 프로세서를 사용하여 네트워크망을 구현하였지만, VMEbus를 통하여 상호 데이터를 주고받고 있기 때문에 외부에서 보기에 하나의 프로세서와 통신을 하는 것과 동일하게 인식한다. 이를 통하여 네트워크망에만 의존하지 않는 이중화 시스템을 적용함으로써 이중화에 대한 안정성과 생존성을 증대시키는 방법을 제안한다.

II. 관련 연구

이중화 기법은 특정 시스템이 고장으로 인해 정지되더라도 다른 시스템이 작업을 이어받아 계속적으로 작업이 가능하도록 하는 것이 주된 목적이다. 이러한 기법은 고가용성을 필요로 하는 시스템에 많이 적용되고 있다. 이중화 기법에는 하드웨어와 소프트웨어를 이용한 다양한 방법이 존재한다[1].

소프트웨어에 대한 이중화 기법으로써 요즘은 네트

워크가 중요한 부분을 차지하는 금융망이나 생산관리, 군사시설 등에서 LAN 스위치 사이에 부가적인 링크를 설치하는 것이 일반적이다. 대표적으로 IEEE 802.1d는 이중화된 구조를 위한 표준안을 제시하고 있다. 그러나 정상적인 상황에서는 단 하나의 활성화된 경로만이 존재하는 것을 전제로 한 규약이므로 로드 밸런싱에 대한 고려가 부족하다.[2].

다른 소프트웨어에 대한 LAN 이중화 초기 구조에서는 정상적인 상태에는 대기 상태로 존재하다가 주 경로에 문제가 발생된 이후에 부 경로가 동작을 시작하는 백업 망의 개념으로 되었다. 이 방법은 가용성의 향상이라는 장점에도 불구하고 네트워크의 속도 향상에는 전혀 기여하지 못하였으며, 네트워크 상태의 지속적 감시를 요구하므로 고가용성이라는 장점에도 불구하고, 평상시에는 오히려 속도를 감소시키는 단점을 지녔다[3].

또한 이중화 기법의 하나로써 프로세서의 두 통신 포트를 별도의 서버에 연결하여 이중화 기능과 모니터링 및 제어 기능을 적용한 방법을 사용한다. 이는 두 개의 서버 즉, 활성 서버와 대기 서버를 적용하여 활성 서버를 통한 통신 수행 시 문제가 발생하였을 경우 이중화 기능을 이용하여 대기 서버로 통신을 지속적으로 수행하기 위한 방법이다[4]. 이러한 기법들이 현재 네트워크의 이중화 구조에 범용적으로 적용되고 있다.

하드웨어 이중화 방식에는 콜드 스탠바이 스페어링, 워 스탠바이 스페어링, 핫 스탠바이 스페어링 기법들이 있다[5]. 콜드 스탠바이 스페어링 기법은 서비스 수행 모듈이 정상적으로 서비스 중인 경우에는 여분의 모듈들은 전원 무공급 상태로 유지하며 현재 서비스 중인 모듈에 고장이 발생하면 여분의 모듈에 전원을 공급하여 서비스를 대신 수행하는 방식이다. 이 기법은 전력 소모를 줄일 수 있는 장점은 있으나 서비스를 정상적으로 수행할 때까지의 시간이 오래 걸리며 하드웨어 고장을 검출하지 못하는 경우가 발생할 수 있다.

워 스탠바이 스페어링 기법은 서비스 수행 모듈이 정상적으로 동작중인 경우에는 콜드 스탠바이 스페어링 기법과는 달리 여분의 모듈들을 전원 공급 상태로 유지시키면서 정상적으로 서비스 중인 수행 모듈의 정보를 부분적으로 동일하게 유지시켜 주는 방식이다. 이 기법은 정상적인 서비스가 수행되기까지의 시간은 단축할 수 있으나 이 방식도 하드웨어 고장을 검출하지 못하는

경우가 발생할 수 있다.

핫 스탠바이 스페어링 기법은 서비스 수행 모듈과 여분의 모듈들을 동시에 전원 공급 상태로 유지시키면서 동일시간 동일동작을 수행하도록 밀결합 동기 상태를 유지하는 방식이다. 이 기법은 서비스 수행중인 모듈 또는 여분의 대기 수행중인 모듈에서 발생할 수 있는 하드웨어의 고장 검출이 가능하므로 시스템의 고가용성 및 고신뢰성을 높일 수 있다. 그러나 이 방식은 다중화 모듈 간의 밀결합 동기과 동일한 정보 자원을 유지해야 하는 어려운 점이 있다.

본 논문에서는 군용 장비의 고가용성 및 고신뢰성을 위하여 핫 스탠바이 스페어링 기법에 해당되는 두 개의 프로세스 간에 VMEbus 통신을 통한 이중화 네트워크 프로토콜망을 구성하고 성능을 확인한다.

III. 프로토콜 설계

3.1 하드웨어 구성

VMEbus를 통한 이중화 네트워크 프로토콜을 수행하기 위한 구성은 그림 1과 같다.

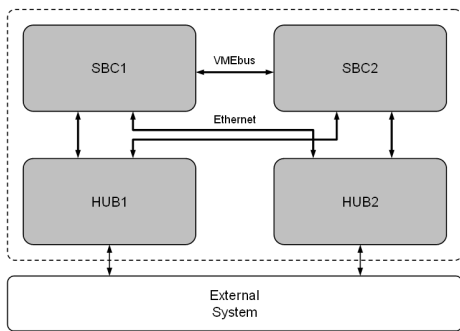


그림 1. 이중화 하드웨어 블록도
Fig. 1 Redundant hardware block diagram

이를 수행하기 위하여 사용하는 프로세스인 두 개의 SBC(Single Board Computer)는 Master 역할과 Slave 역할로써 상호 간에 VMEbus 통신을 통하여 데이터를 주고받는다. 그리고 각각의 SBC는 두 개의 Ethernet 네트워크망이 두 개의 HUB로 연결되며, 각각의 채널 1은 HUB1, 그리고 채널 2는 HUB2로 연결된다. 외부 장

비와의 네트워크망은 각각의 HUB에서 하나의 채널이 나와 최종적으로 두 개의 망으로 연결된다. 이로써 내부에서는 두 개의 SBC를 사용하여 네트워크망을 구현하였지만, VMEbus를 통하여 상호 데이터를 주고 받고 있기 때문에 외부에서 보기에는 하나의 프로세서와 통신을 하는 것과 동일하게 인식하며, 전체 4개의 채널을 통하여 통신을 수행하여 생존성을 증가시킬 수 있다.

Master 역할의 SBC1과 Slave 역할의 SBC2는 각각 우선한 채널씩 외부와 실시간 통신을 수행하며, VMEbus를 통하여 상호 Ethernet 네트워크망과 SBC 자체 상태를 주고받게 된다.

3.2 소프트웨어 구성

소프트웨어는 RTOS(Real Time Operating System)인 VxWorks를 통하여 구현하였다[6].

이중화 네트워크 프로토콜은 VMEbus Master 역할의 SBC1과 Slave 역할의 SBC2 간의 공유 메모리 영역과 인터럽트 방식을 이용하여 구현한다. 그러기 위하여 이중화를 담당하는 전용 Task인 tManagerTask가 존재한다.

3.2.1 tManagerTask의 기능

tManagerTask의 기능은 그림 2와 같다.

tManagerTask는 ‘등록된 이벤트 감시’와 ‘이벤트 처리’를 수행하는 두 가지 기능으로 나눌 수 있다. ‘등록된 이벤트 감시’ 기능은 먼저 ‘SBC 상태 감시’ 기능을 통하여 주기적으로 SBC의 상태를 감시한다. 이 기능을 통하여 SBC의 부분적인 통신 기능의 손상뿐만 아니라 완전 손상에 대비하여 다른 SBC가 모든 기능을 상속받을 수 있게 한다.

두 번째로 ‘LAN1 통신 감시’ 기능을 통하여 Ethernet LAN1 통신 기능에 이상이 발생한 경우에 이벤트를 발생시킨다. 세 번째로 ‘LAN1 통신 감시’ 기능을 통하여 Ethernet LAN2 통신 기능에 이상이 발생한 경우에 이벤트를 발생시킨다. 그 밖의 통신 기능에 대비하여 ‘예비’ 기능으로 포함하였고, 마지막으로 ‘인터럽트 감시’ 기능을 통하여 SBC의 상태, Ethernet LAN1 통신 상태, 그리고 Ethernet LAN2 통신 상태를 알리기 위하여 VMEbus 공유 메모리에 Write하고 발생한 인터럽트를 이벤트로 등록하여 처리하게 된다.

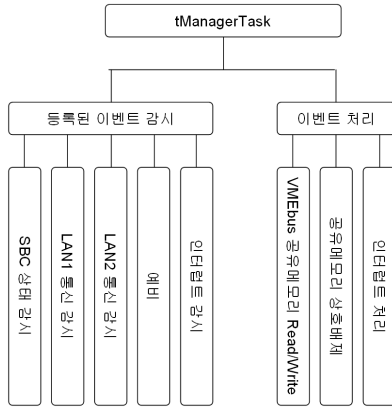


그림 2. tManagerTask의 기능
Fig. 2 tManagerTask features

tManagerTask의 두 번째 기능인 ‘이벤트 처리’에 대한 첫 번째 ‘VMEbus 공유 메모리 Read/Write’ 기능은 두 개의 SBC 간에 정보를 공유하게 되고, 두 번째 ‘공유메모리 상호배제’ 기능은 VMEbus의 Read/Write 수행 시 정보가 손상되는 것을 막아준다. 마지막으로 ‘인터럽트 처리’ 기능은 VMEbus의 공유 메모리 영역에 Write 한 후 인터럽트를 발생시켜 다른 SBC에 알려주는 기능을 갖는다.

외부 장비와의 통신을 담당하는 Task에서 통신상의 이상이 발생하면 tManagerTask에게 이벤트를 발생시켜 이상이 발생하였음을 알려준다. 그리고 이벤트 종류에 따라 VMEbus의 Master와 Slave 간의 공유 메모리 영역에 데이터를 Write 한다. 그리고 Master 역할의 SBC1은 인터럽트를 발생시켜 통신 이상이 발생하였음을 Slave 역할의 SBC2에게 알려준다. Slave 역할의 SBC는 인터럽트가 수신된 것을 확인한 후 VME Master와 Slave 간의 공유 메모리 영역으로부터 데이터를 Read한다. 그리고 Read한 데이터의 내용에 따라 해당 통신 채널을 오픈한다.

3.2.2 이중화 시퀀스

VMEbus 이중화 네트워크 프로토콜의 정상동작 시퀀스 다이어그램은 그림 3과 같다.

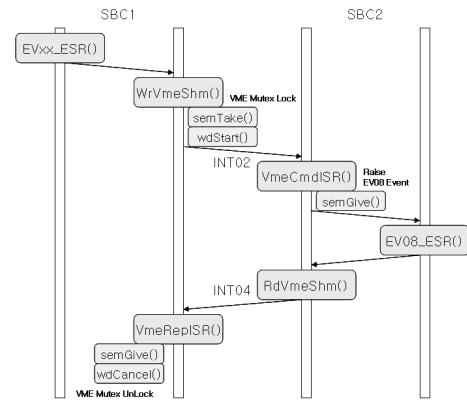


그림 3. 이중화 시퀀스 다이어그램
Fig. 3 Redundant sequence diagram

프로토콜을 위해 구현한 함수로는 이벤트 처리 서비스 루틴 함수와 VMEbus Read/Write 함수, 그리고 인터럽트 처리 함수로 나눌 수 있다. 이벤트 처리 서비스 루틴 함수는 EVxx_ESR() 함수로서 각 이벤트를 처리하는 함수이다. VMEbus Read/Write 함수로는 공유 메모리에 Write를 수행하는 기능의 WrVmeShm() 함수와, Read를 수행하는 기능의 RdVmeShm() 함수가 있다. 마지막으로 인터럽트 처리 함수로는 VmeCmdISR() 함수와 VmeReplISR() 함수가 있다.

기존에 VxWorks 라이브러리로 제공되는 함수로는 WatchDog 제어 함수와 세마포어 제어 함수를 사용하였다. WatchDog 제어 함수에는 WatchDog 기능을 시작하는 wdStart() 함수와 중지하는 기능의 wdCancel() 함수를 사용하였고, 공유 메모리 상호배제 기능을 위하여 세마포어 제어 함수를 사용하였으며, semTake() 함수와 semGive() 함수가 있다.

시퀀스의 진행은 다음과 같다. 먼저 SBC1의 상태 불량, Ethernet LAN1의 통신 상태 불량, Ethernet LAN2의 통신 불량 중 하나의 이벤트가 발생하면 이벤트 서비스 루틴 함수인 EVxx_ESR() 함수를 수행한다. SBC1은 SBC2에게 이 상태를 알리기 위하여 VMEbus 상의 공유 메모리에 데이터를 쓰기 위한 WrVmeShm()를 수행한다. 그리고 공유 메모리 상호배제 기능을 위하여 세마포어 제어 함수 semTake()를 실행하고, VxWorks 라이브러리로 제공되는 함수인 WatchDog 제어 함수로 WatchDog 기능을 시작하는 wdStart() 함수가 수행된다.

해당하는 인터럽트 INT02가 발생하면 SBC2에서는 인터럽트 처리 함수인 VmeCmdISR() 함수가 수행한다. 이로써 공유 메모리 상호배제 기능을 위하여 세마포어 제어 함수 semGive() 함수가 수행하고, 이벤트 EV08을 발생시킨다. 그러면 해당하는 이벤트 서비스 루틴 함수 EV08_ESR()가 발생한다. 이에 따라 공유 메모리로부터 읽기를 수행하는 기능의 RdVmeShm() 함수가 수행한다. 수행 후 SBC2가 SBC1로 인터럽트 INT04를 발생시키면 마지막으로 SBC1에서는 인터럽트 처리 함수 VmeRepISR() 함수가 수행한다. 이후 공유 메모리 상호배제 기능을 위하여 세마포어 제어 함수 semGive() 함수와 WatchDog 기능을 중지하는 기능의 wdCancel() 함수가 수행하며 시퀀스가 종료한다.

3.2.3 이중화 Task 기능

Ethernet 통신 프로토콜을 운용하기 위하여 본 시스템은 이중화를 담당하는 전용 Task인 tManagerTask 외에 여러 Task를 필요로 하며 각각의 기능은 표 1과 같다.

표 1. 이중화 Task 기능
Table. 1 Redundant task features

Task명	기능
tLan1UdpRcvTask	LAN1 UDP 메시지 수신
tLan2UdpRcvTask	LAN2 UDP 메시지 수신
tMsgFilterTask	중복 메시지 분류
tDataProcTask	수신 메시지 처리
tUdpSendTask	UDP 메시지 송신

tLan1UdpRcvTask와 tLan2UdpRcvTask에서 UDP 통신을 통하여 외부 장비로부터 메시지를 수신하면 tMsgFilterTask에서 중복 메시지 여부를 판별하여 중복 메시지인 경우 메시지를 버린다. tDataProcTask에서 중복 메시지가 아닌 경우에는 수신 메시지를 처리한 후 메시지 처리 결과를 tUdpSendTask를 통하여 외부 장비에 송신한다.

IV. 구현 및 결과

VMEbus를 통한 이중화 네트워크 프로토콜을 구현하여 확인한 화면과 하드웨어 구성은 그림 4와 같다. SBC1

의 Ethernet LAN1의 통신 기능에 이상이 발생한 후 이중화가 수행하여 다른 채널로 통신망이 이동한 결과의 화면은 그림 5와 같다. LAN1은 SBC1의 Ethernet LAN1 채널을 의미한다. 그리고 LAN2는 SBC2의 Ethernet LAN2 채널을 의미한다. 오른쪽은 외부 장비를 의미한다. 통신이 정상적으로 수행하여 모든 상태가 정상으로 표시된다. 아래의 연결도에서 SBC1의 Ethernet LAN1 채널과 SBC2의 Ethernet LAN2 채널이 외부 장비의 두 채널과 네트워크망을 이루어 연결된 연결도를 확인할 수 있으며, 통신은 실시간으로 지속되는 상태로 볼 수 있다.

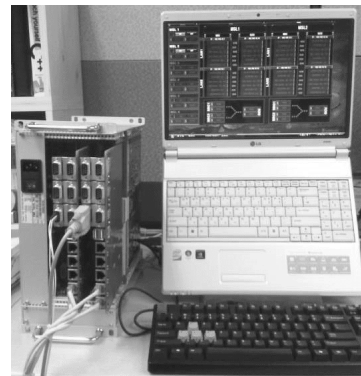


그림 4. 하드웨어 구성
Fig. 4 Hardware structure

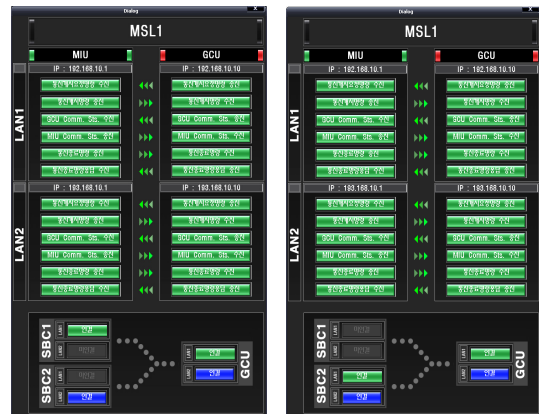


그림 5. 이중화 수행 전/후 연결도
Fig. 5 Connection before/after performing redundancy

통신 수행 중 SBC1의 Ethernet LAN1의 통신 기능에 이상이 발생한 경우 SBC1의 Ethernet LAN1 채널은 현재

통신 중인 SBC2의 Ethernet LAN1 채널과 동일한 네트워크망이므로 SBC2의 Ethernet LAN1 채널이 VMEbus를 통한 상태를 주고받는 결과로 통신권을 실시간으로 이어받아 통신이 정상적으로 수행하게 되며, 모든 상태가 정상으로 표시된다. 아래의 연결도에서 SBC2의 Ethernet LAN1 채널과 Ethernet LAN2 채널이 외부 장비의 두 채널과 네트워크망을 이루어 연결된 연결도를 확인할 수 있으며, 통신은 실시간으로 지속되는 상태로 볼 수 있다.

이와 같은 실험을 통하여 두 SBC 각각의 비정상 동작을 모의하고, Ethernet LAN1과 Ethernet LAN2의 통신 기능에 이상이 발생한 경우를 모의하여 다른 여유 채널을 통한 외부 장비와의 실시간 통신이 유지되는 것을 확인함으로써 본 논문에서 제안한 프로토콜에 대한 안정성과 생존에 대한 성능을 확인하였다.

V. 결 론

본 논문을 통하여 향상된 이중화 설계 기법을 제시했으며 기존의 설계 방식보다 시스템의 안정성과 생존성을 높일 수 있음을 실험을 통해 확인했다. 향후 개발하는 군용 장비의 네트워크에는 VMEbus 통신을 통한 이중화 네트워크 프로토콜을 적용하여 좀 더 안정적이고 생존성이 높아진 군용 장비를 개발할 수 있을 것으로 기대한다.

참고문헌

[1] 이재승, 류준하, “전투체계 시스템을 위한 실시간 환경에서의 비동기 이중화 기법 연구”, 한국군사과학기술학회지 제10권 제2호, pp.61-68, 2007.

[2] 박지훈, 박종규, 한일석, 김학배, “서버 독립적 LAN 이중화 모듈을 통한 네트워크 속도 및 가용성 향상 방안”, Journal of Control, Automation and Systems Engineering. Vol. 8. pp.264-269, 2002.

[3] 박지훈, 한일석, 김학배, “네트워크 속도 향상과 고 가용성 보장을 위한 서버 독립적 LAN 이중화 방법”, 대한전기학회 하계학술대회 논문집, pp.2711-2713, 2001.

[4] 류길수, “감시제어 시스템의 신뢰성 향상을 위한 서버 및 네트워크 이중화”, 한국박용기관학회 춘계학술대회논문집, pp. 349-357, 2004.

[5] 신진욱, 박동선, “핫 스탠바이 스페어링 기법을 이용한 고장 감내 이중화 시스템 설계”, 한국통신학회논문지 '04-10 Vol.29 No.10A, pp.1113-1122, 2004

[6] <http://ko.wikipedia.org/wiki/VxWorks>



박정원(Jeong-Weon Park)

1996년 홍익대학교 전파공학과 공학학사

2010년 아주대학교 전자공학과 공학석사

1996년~현재 LIG넥스원(주) 수석연구원

※관심분야: 지휘통제, 전투체계 통신, System Engineering



박성진(Seong-Jin Park)

1994년 KAIST 전기및전자공학과 공학학사

1997년 동대학원 공학석사

2001년 동대학원 공학박사

2001~2004년 삼성전자 네트워크사업부 책임연구원

2004년~현재 아주대학교 전자공학부 교수

※관심분야: Discrete event systems