

안전성과 신뢰성을 위한 선박 LAN 구축 방안

김영수* · 조익성* · 임재홍**

A Ship's LAN Configuration Method for the Safety and Reliability

Yeong-Su Kim · Ik-Sung Cho* · Jae-Hong Yim***

Abstract	4. 시험 및 고찰
1. 서론	5. 결론
2. 신뢰성을 위한 고장허용 기법	참고문헌
3. 신뢰성 있는 네트워크 구성 방안	

Abstract

As a shipboard dependency for the safety and reliability becomes very important, the need for solid systems providing non-stop workload has been increased. This system is heartbeat that transmits shipboard state, audit and control information to the land. So, this paper describes a ship's LAN configuration method for the safety and reliability. In order to achieve these requirements, network, server and disk fault tolerance techniques are surveyed, and dual network configuration model, cluster server configuration method are presented and tested based on the survey.

1. 서론

네트워크가 일반화되지 않았을 당시에 기업들은 필요한 모든 정보를 하나의 시스템에서 관리하는 중앙집중식 관리방식을 취하였다. 이러한 방식은 정보를 구축하는 측면과 운영하는 측면에서

많은 비용이 필요하고, 컴퓨터 시스템을 새로운 시스템으로 바꾸는 과정에서도 많은 문제점을 안고있어 네트워크를 통한 정보의 공유 및 분산 컴퓨팅 환경으로의 발전이 가속화되고 있는 추세이다. 또한 선박 자동화 시스템 기술의 발달로 선박 내에서도 LAN(Local Area Network)은 물론

* 본 논문은 산업자원부 중기저점과제 "선박용 항해·통신장비 개발사업" 중 "종합항법장치 SI 기술개발" 과제의 일부로 수행되었음

* 한국해양대학교 전자통신공학과 박사과정

** 한국해양대학교 전자통신공학과 조교수

다양한 데이터의 효율적인 관리를 위한 데이터베이스 구축이 늘어나고 있는 추세이며, 선박내의 현재 상태나 제어정보는 항해중인 선박의 안전과 더불어 육상과의 정보교환에도 중요한 위치를 차지하고 있다. 이에 따라 선박 내 네트워크 환경에서는 육상에서의 환경과는 달리 안전성과 신뢰성을 고려하여 장애에 대비할 새로운 방법이 요구되고 있다¹⁾.

신뢰성 있는 컴퓨팅 환경을 구성하기 위해서는 구체적으로 서버 시스템 및 네트워크, 저장장치 등의 이중화를 통하여 장애에 대비하는 하드웨어적인 준비가 필요하며, 소프트웨어적으로는 복수의 서버를 기억하고 있다가 서비스가 가능한 서버를 찾아 요청을 보내도록 클라이언트 프로그램을 개발하거나 미들웨어(middleware)를 사용해야만 한다²⁾.

본 논문에서는 선박의 안전성과 신뢰성을 위한 선박 LAN 구축을 위하여 네트워크, 서버 및 디스크의 고장허용(fault tolerance) 기법을 고찰하고, 이를 토대로 단일 네트워크의 문제점을 해결하기 위한 이중 네트워크의 구성, 데이터베이스 및 주요 서버 프로그램을 관리하는 서버의 장애를 해결한 클러스터(cluster) 서버 구성 등 신뢰성 있는 선박 LAN 구축 방안을 제안한다. 또한 제안한 방법대로 실제 테스트 LAN을 구성하고 네트워크 및 서버의 장애 시험을 통하여 신뢰성을 확인하도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 신뢰성을 위한 고장허용(fault tolerance) 기법의 개요에 대하여 논하고, 3장에서는 신뢰성 있는 네트워크 구성 방법에 대하여, 4장에서는 시험 및 고찰에 대하여, 마지막으로 5장에서는 결론을 서술한다.

2. 신뢰성을 위한 고장허용 기법

2.1 네트워크 고장허용 기법

LAN의 구성 방식에는 IEEE(Institute of Electrical and Electronic Engineers)의 표준인 이더넷(ethernet), 토큰 버스(token bus), 토큰 링(token ring) 등과 ANSI(American National Standards Institute) 표준인 FDDI(Fiber Distributed Data Interface) 등 여러 방식이 있다³⁾. 이더넷과 토큰 버스는 버스형 접

속형태(topology)를 사용하고, 토큰 링과 FDDI는 링형 접속형태를 사용하고 있다. 본 장에서는 접속형태별로 네트워크 고장허용 방안을 고찰하도록 한다.

2.1.1 버스형 접속형태

버스형 접속형태에서는 이더넷 방식과 토큰 버스 방식이 사용될 수 있는데, 토큰 버스 방식은 공장 자동화 분야에 사용되고, LAN에서는 주로 이더넷 방식이 사용되므로 이더넷 방식에 대해서만 논하도록 한다. 이더넷의 전송 방식은 CSMA/CD(Carrier Sense Multiple Access with Collision Detection)로써 경쟁 방식을 취하고 있다. 즉, 전송매체를 통하여 전송되고 있는 데이터가 없으면 어느 노드나 데이터 전송이 가능하다.

이더넷은 버스형과 더불어 허브(hub)를 사용하는 스타형 접속형태도 가능하나 전송 메커니즘은 동일하며, 이러한 이더넷 환경에서는 네트워크의 문제 발생시 전체 네트워크 동작이 마비되므로 네트워크 고장허용을 위해서는 그림 1과 같이 이중 네트워크를 구성하는 수밖에 없다. 또한, 이중 네트워크를 사용할 경우 고장허용의 정상적인 동작을 위해서는 네트워크의 상태를 계속 모니터링 하여 메인 네트워크에 이상이 있을 경우 백업 네트워크

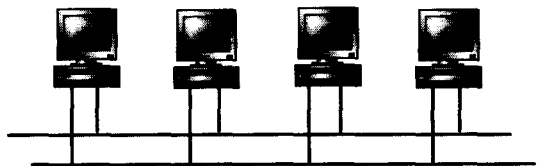


그림 1. 이중 네트워크

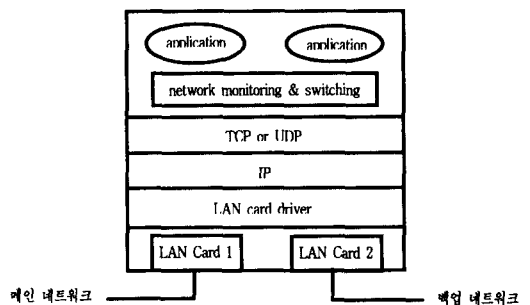


그림 2. 네트워크 모니터링 및 스위칭

크로 스위칭 되어 데이터를 전송할 수 있도록 이더넷 상위 계층에서 그림 2와 같이 모니터링 및 스위칭 프로그램이 수행되어야 한다. 이더넷 방식에서는 어느 한 노드에 이상이 발생했을 경우 전체 네트워크에는 영향을 미치지 않고, 문제가 발생한 노드만 네트워크에서 제외된다.

2.1.2 링형 접속형태

링형 접속형태에서는 토큰 링과 FDDI 방식이 사용될 수 있는데, 두 방식 모두 토큰을 수신한 노드에게만 데이터 전송 권한을 부여하는 토큰 패싱(token passing) 전송 방식을 사용하고 있다.

토큰 링에서는 MAU(Multistation Access Unit)라고 하는 허브를 통하여 링을 구성하고 각 노드의 증계 기능에 의하여 데이터 및 토큰을 전송하게 된다⁴⁾. 따라서, 노드나 네트워크에 문제가 발생할 경우 전체 네트워크가 마비되지만, 최근에는 MAU 내에 자동 교환기(automatic switch)가 설치되어 노드가 동작하지 않을 경우 그 노드를 링에서 제외시키고 정상적인 링으로 동작하게 한다. 하지만 네트워크에 이상이 있을 경우에는 여전히 전체 네트워크가 마비되므로 이를 해결하기 위해서는 이더넷 방식에서와 마찬가지로 이중 링을 구성하고 네트워크 스위칭 프로그램을 이용해야만 한다.

FDDI는 광케이블을 사용하는 링 방식의 고속 네트워크로서 최고 100Mbps의 전송 속도를 제공한다. 토큰 링의 단일 링 문제점을 보완한 FDDI에서는 그림 3(a)와 같이 이중 링 구조로 구성되어 정상 시에는 1차 링(primary ring)을 통하여 데이터 전송을 수행하고, 1차 링에 문제가 발생할 경우 그림 3(b)와 같이 2차 링(secondary ring)을 이용하여 링을 재구성함으로써 정상 동작을 유지하게 된다⁴⁾. 이러한 구성은 FDDI로 하여금 고장이 발생하더라도 링 동작을 계속 유지하면서 고장난 요소를 제거시키고 링을 재구성할 수 있도록 한다. 링은 고장난 요소를 계속 관찰하고 있다가 복구가 완료되면 다시 링을 원상태로 재구성하여 정상적인 동작으로 되돌려 놓는다. 따라서 FDDI는 자체적으로 고장허용 기능을 제공하기 때문에 주로 백본(backbone) 네트워크를 구성하는데 많이 이용되고 있다.

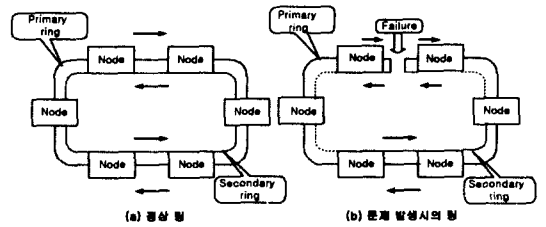


그림 3. FDDI 링 구조

2.2 서버 고장허용 기법

데이터베이스 서비스, 파일 서비스, 프린트 서비스 등을 제공하는 서버의 장애 역시 LAN 환경에서는 치명적인 결과를 초래하게 된다. 따라서 안전성과 신뢰성을 위해서는 서버의 고장허용이 제공되어야 하는데, 서버의 고장허용을 위한 여러 방식 중 가장 널리 사용되는 무정지 시스템(non-stop system)과 클러스터(cluster) 방식에 대해서 논하도록 한다.

2.2.1 무정지 시스템

무정지 시스템은 메인 서버와 백업 서버로 서버 시스템을 이중화하여, 그림 4와 같이 메인 서버에 장애가 발생할 경우 백업 서버가 메인 서버에서 수행 중이던 프로세스를 인계 받아 메인 서버의 역할을 대신 수행하도록 하는 기법이다. 이 방식은 메인 시스템과 이 시스템이 정지했을 때를 대비하여 항상 대기상태로 있는 백업 시스템으로 구성되며, 두 시스템은 항상 모든 동작 상태를 일치시켜 메인 시스템 장애시 백업 시스템이 계속 동작을 이어가도록 하는 방법이다. 그러나 이 방식에서는 메인 서버가 정상적인 경우 백업 서버는 유휴 상태에 있게 되므로 자원의 낭비가 크다는 단점이 있다.

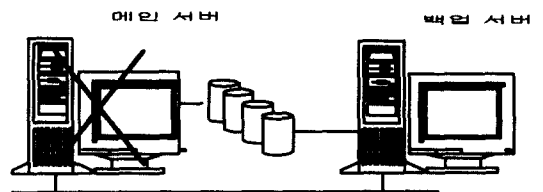


그림 4. 무정지 시스템

무정지 시스템 방식은 99% 이상의 업 타임(up-time)을 보장한다는 장점이 있지만 비용이 많이 들고 구현이 까다로우며 시스템의 가동 효율성이 낮고 시스템의 기종 선택에 있어서도 제한이 많이 따르게 된다⁵⁾.

2.2.2 클러스터

클러스터란 그림 5와 같이 느슨하게 연결된(loosely coupled) 2개 이상의 시스템과 이들 시스템이 공유하는 디스크로 구성되어 마치 하나의 서버처럼 동작하게 하는 것으로, 컴퓨팅 차원에서 고도의 가용성과 확장성 및 유연성, 관리의 편의성을 제공하는 서버의 고장허용 기법이다⁵⁶⁾. 클라이언트 시스템은 둘 이상의 시스템으로 구성되는 클러스터 서버를 하나의 서버로 인식하여 상호 동작하게 된다. 클러스터의 실제 구성은 두 시스템을 SCSI(Small Computer System Interface) 버스를 공유하는 방식으로 결합하여 클러스터 또는 단일 시스템 환경을 구축하고, 클라이언트는 클러스터 내의 어떤 서버를 사용하는지 인식할 필요 없이 클러스터의 모든 자원(디스크, 파일, 데이터베이스 등)에 액세스 할 수 있다.

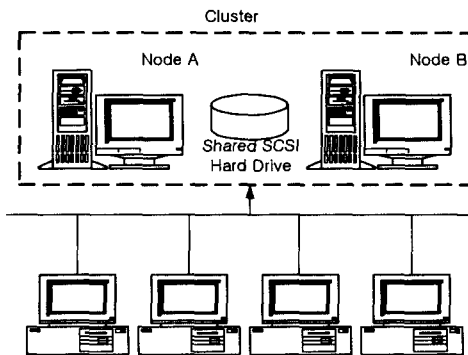


그림 5. 클러스터

클러스터가 제공하는 가용성 수준은 무정지 시스템과 비교되는데, 클러스터는 한 서버의 완벽하게 미러(mirror)된 백업을 필요로 하지 않기 때문에 비용 면에서 매우 효과적이다. 즉, 무정지 시스템은 가용성을 제공하지만 백업 시스템의 비용을

요구하는 반면, 클러스터는 사용자가 작업분산(workload)을 할당하고, 두 서버를 사용하는 것을 허용한다. 따라서 사용자는 더 큰 용량의 이익을 갖고, 현 자원에의 투자를 극대화 할 수 있다.

2.3 디스크 고장허용 기법

네트워크의 문제를 방지하는 대책의 하나로써 시스템의 디스크를 오류 지향성 디스크로 설치하는 방법이 있다. 오류 지향성 시스템은 여러 요소로 이루어져 하나의 요소가 문제를 일으켜도 다른 요소에서 동일한 기능을 수행하는 것을 기본원리로 한다. 오류 지향성 디스크 시스템도 하드웨어적인 것으로 일부 소프트웨어 요소를 포함하여 네트워크의 저장 장치의 오류에 의한 문제를 예방하는데 사용된다. 오류 지향성 디스크 시스템의 기본적인 방식은 RAID(Redundant Arrays of Inexpensive Disk) 등급을 사용한다. 표 1은 RAID 등급의 규격을 표시한다⁷⁾.

표 1. RAID 등급

RAID 등급	적용
0	데이터 블록(block)을 사용하여 여러 디스크에 기록
1	디스크의 복제 및 복사를 실행
2	오류 수정 코드로서 비트 크기로 여러 디스크에 기록
3	한 개의 드라이브에 패리티 정보를 바이트 크기로 기록
4	한 개의 드라이브에 패리티 정보를 블록 단위로 기록
5	여러 디스크에 패리티 정보를 블록 단위로 기록

RAID 시스템은 디스크의 조각(stripe)을 기본으로 사용한다. 이는 여러 개의 디스크에 동일한 데이터를 기록하는 것을 말하며 하나의 디스크 제어기는 여러 디스크를 제어하여 데이터의 개별 바이트 또는 데이터 블록(block) 단위로 기록한다. 이러한 디스크 시스템은 대부분의 네트워크에서 시스템의 부팅 기능을 갖지 않으며 단순한 저장의 기능

만을 갖는다.

마이크로소프트 윈도우 NT의 경우는 RAID 등급 0, 1 그리고 5를 지원하는데 이는 디스크 관리자 도구에서 실행된다. RAID 등급 5는 RAID 등급 2와 3, 그리고 4로부터 생긴 것이므로 최근 기법인 등급 5만을 지원한다⁷⁾.

3. 신뢰성 있는 네트워크 구성 방안

3.1 전체 네트워크 구성

신뢰성 있는 선박 LAN 구축을 위한 전체 네트워크 구성도는 그림 6과 같다.

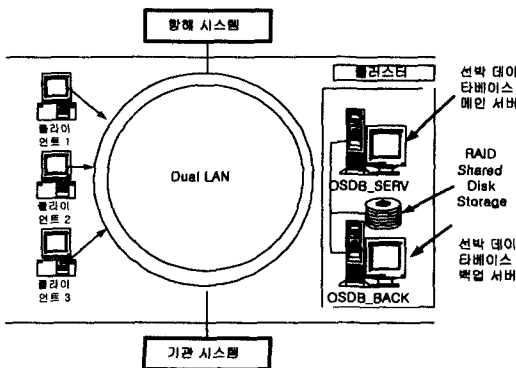


그림 6. 전체 네트워크 구성도

그림에서와 같이 선박 LAN은 항해 자동화 시스템, 기관 자동화 시스템, 데이터베이스 서버 및 여러 클라이언트로 구성될 수 있으며, 네트워크의 고장허용을 위하여 이중 네트워크로 구성하게 된다. 서버의 고장허용을 위해서는 클러스터 서버 방식을 택하고, 디스크는 미러링이 제공되는 RAID 디스크를 사용하게 된다.

본 논문에서는 네트워크 상위 프로토콜로써 가장 많이 사용되는 TCP/IP(Transmission Control Protocol/Internet Protocol) 프로토콜을 사용하는 것으로 가정한다. TCP/IP 프로토콜을 택한 이유는 이더넷, 토큰 링, FDDI, ATM(Asynchronous Transfer Mode) 등 네트워크의 종류에 관계없이 사용할 수 있고, 마이크로소프트 네트워크 환경을

사용할 경우에도 NBT(NetBIOS over TCP/IP)를 통한 전송이 가능하여 이식성 및 유연성이 매우 우수하기 때문이다.

3.2 네트워크 고장허용 구현

네트워크 장애는 네트워크 인터페이스 카드(NIC: Network Interface Card), 네트워크 케이블, 네트워크 중계기(라우터, 허브 등) 등에서 발생할 소지가 많으며, 네트워크 장애시 발생부위를 찾아내기 위해서는 많은 시간이 소요된다. 그러므로 네트워크의 장애시간을 최소화하기 위하여 케이블을 이중화하고 각 시스템에 LAN 카드를 2개씩 장착하여 이중 네트워크를 구성하도록 한다. 또한 각 컴퓨터에는 이중 네트워크를 위한 LAN 카드 각각에 IP 주소가 할당되어야 한다. 따라서, 각 컴퓨터는 호스트 이름을 통하여 서버에 접속할 때 메인 네트워크에 이상이 없는 경우는 메인 IP 주소를 사용해야 하고 메인 네트워크에 이상이 있는 경우는 백업 IP 주소를 통한 백업 네트워크를 이용하여 서버에 접속하게 된다. 이와 같이 동작하기 위해서는 2.1절에서도 언급하였듯이 네트워크 스위칭 프로그램이 필요하며, 스위칭 프로그램 구현의 용이함을 위하여 먼저 TCP/IP 환경에서 호스트 이름 분해 순서를 살펴보면 그림 7과 같다.

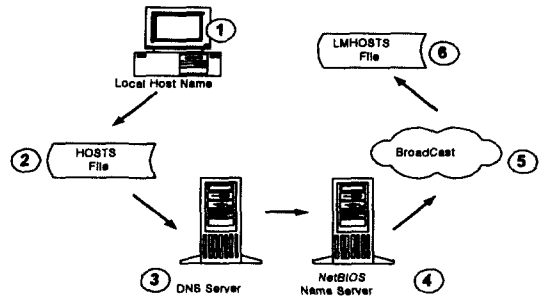


그림 7. TCP/IP 호스트 이름 분해 절차

그림 7에서와 같이 사용자가 호스트 이름을 참조하는 명령을 입력하면, 해당 호스트 이름이 로컬 호스트 이름과 동일한지 확인하고, 호스트 이름과 로컬 호스트 이름이 다르면, "HOSTS" 파일이 파

성된다. "HOSTS" 파일에서 이름을 찾은 경우에는 IP 주소로 분해하여 접속하고, "HOSTS" 파일에서 호스트 이름을 찾지 못한 경우 DNS(Domain Name System) 서버에 요청하게 된다. DNS 서버에서 찾지 못할 경우, NetBIOS 이름 서버(WINS 서버)에 요청하게 되고 NetBIOS 이름 서버에서도 찾지 못할 경우, 로컬 네트워크에 방송(broadcast)하여 호스트 이름을 탐색하게 된다. 그래도 찾지 못할 경우에는 마지막으로 "LMHOSTS" 파일을 검색하게 된다. 위 방법 중 어느 하나도 호스트 이름을 찾지 못하는 경우에는 IP 주소로 직접 접속해야 한다.

본 논문에서는 네트워크 스위칭 프로그램의 구현을 위해서 "HOSTS" 파일을 이용하도록 한다. 그 이유는 선박 LAN은 소규모 네트워크이므로 DNS 서버나 WINS 서버가 없을 수 있고, 가장 간단하게 구성할 수 있기 때문이다. "LMHOSTS" 파일을 사용해도 무방하나 호스트 이름 분해 절차상

"HOSTS" 파일을 이용하는 것이 분해 속도를 빠르게 할 수 있다는 장점이 있다. 네트워크 스위칭 프로그램을 위한 순서도는 그림 8과 같다.

3.3 클러스터 인식 프로그램 구현

본 논문에서는 서버의 고장허용을 위해서 시스템의 효율성이 높은 클러스터 서버 방식을 택하고, 디스크는 미러링이 제공되는 RAID 디스크를 사용한다. 그러나 클러스터 서버를 사용한다고 해도 서버 내에서 사용되는 프로그램이 클러스터 기능을 제공하지는 않는다. 프로그램이 클러스터 기능을 제공하기 위해서는 클러스터를 인식할 수 있어야 하는데, 본 절에서는 클러스터 인식 프로그램 구현 기법에 대하여 기술한다.

3.3.1 클러스터 서버의 구조

클러스터에 대한 인터페이스를 제공하는 소프트웨어 컴포넌트는 클러스터 서비스, 클러스터 디스크 드라이브, 클러스터 네트워크 드라이브, 리소스(resource) 모니터, 리소스 DLL(Dynamic Link Library), 클러스터 관리자, 클러스터 API(Application Programming Interface), 클러스터 자동화 서버, 클러스터 데이터베이스로 나뉘어지며, 각 컴포넌트는 클러스터를 수행하기 위하여 함께 동작한다. 그림 9는 마이크로소프트 윈도우즈 NT 서버 운영체제에서 클러스터 컴포넌트들간의 관련성과 동작구조를 나타내고 있다.

클러스터를 지원하는 응용 프로그램은 클러스터 API를 통하여 클러스터 서버에 의해 제공되는 특징을 이용할 수 있다. 클러스터를 지원하는 응용 프로그램은 리소스 모니터에 의해 요청된 상태를 보고하며, 성공적으로 온라인(online) 또는 오프라인(offline) 되었는지 요청에 응답한다. 클러스터를 지원하는 응용 프로그램은 응용 프로그램을 책임지고 있는 개발자들에 의해 만들어진 리소스 유형인 커스텀 자원유형으로서 클러스터 서비스에 의해 관리될 수 있다. 커스텀 리소스 유형을 만들기 위해서는 개발자들은 두 가지 DLL을 만들어야만 한다. 즉 리소스 DLL과 클러스터 관리 확장 DLL이다^{8,9)}. 그림 10은 여러 종류의 리소스와 리소스

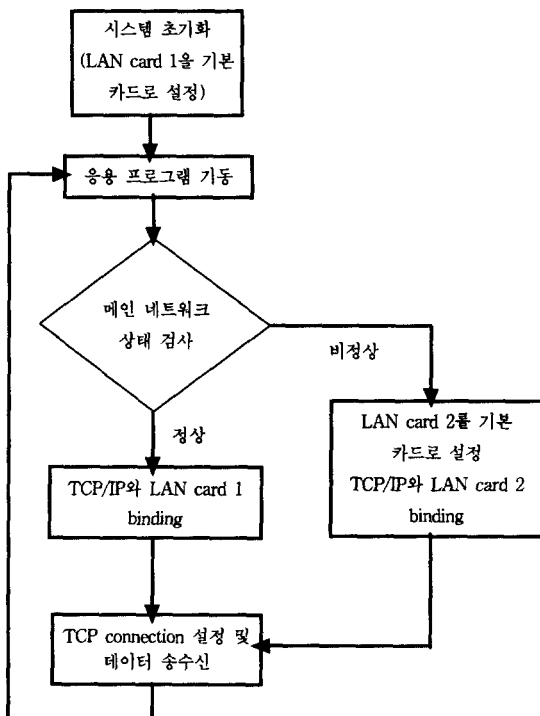


그림 8. 네트워크 스위칭 프로그램 순서도

DLL과의 관계를 나타낸다.

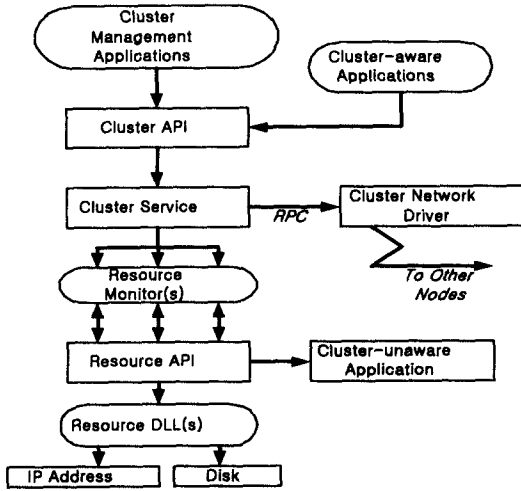


그림 9. 클러스터 서버의 구조

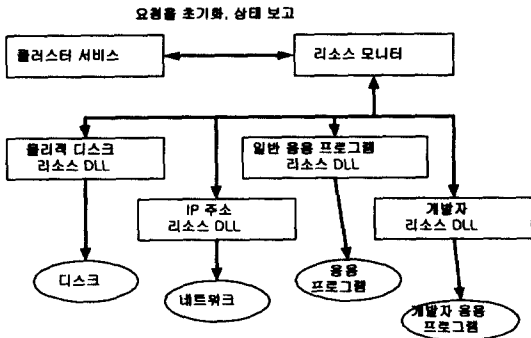


그림 10. 리소스 DLL과 리소스

3.3.2 구현 방법

클러스터 인식 프로그램의 구현 절차는 다음과 같다¹⁰⁾.

- (1) 클러스터를 지원할 응용 프로그램을 작성한다.
- (2) 클러스터 서버가 응용 프로그램과 서비스를 관리하도록 새로운 클러스터 리소스 DLL을 만든다.
- (3) 클러스터 서버가 커스텀 리소스 유형을 조정하도록 클러스터 관리 확장 DLL을 만든다.
- (4) 리소스가 클러스터를 인식할 수 있도록 리소스 DLL을 클러스터 서버에 등록한다.

(5) 클러스터의 동작여부를 확인하기 위하여 작업인계를 시험한다.

리소스 DLL과 관리 확장 DLL을 만들기 위해서는 비주얼 도구에서 제공하는 리소스 유형 마법사 프로그램을 이용하면 된다. 새로운 리소스 유형을 만들 때 클러스터 관리자는 클라이언트가 접근하는 가상서버를 구성할 네트워크 이름과, 그것과 관계된 리소스를 새로운 유형과 결합시켜야 한다. 네트워크 관리자가 그룹을 형성할 때, 필요한 리소스들에 대한 의존성이 결여되었다면 클러스터 관리 확장 DLL의 상태를 점검하고, 리소스 DLL을 적재함으로써 같은 그룹 내의 필요한 모든 의존성을 위치시켜야 한다. 예를 들면, 새로운 리소스 유형이 데이터 파일에 관련된 클러스터 응용 프로그램을 관리한다고 가정하면, 관리자는 이 응용 프로그램과 응용 프로그램의 데이터 파일, 관련된 네트워크 네임 리소스, IP 주소 리소스, 공유 디스크 리소스를 형성해야 한다⁸⁾.

4. 시험 및 고찰

네트워크 및 서버의 장애 시험을 위하여 그림 11과 같이 시험용 LAN을 구성하였다. 네트워크 종류의 투명성을 확인하기 위해서 메인 네트워크는 155 Mbps ATM 네트워크로 구성하고, 백업 네트워크는 100 Mbps 이더넷 네트워크로 구성하였다. 또한 클러스터 서버를 구성하는 두 대의 서버는 별도의 네트워크를 갖추게 함으로써 실제 데이터 전송을 위한 네트워크의 부하에는 전혀 영향을 주지 않는다. 그림 11과 같이 이중 네트워크를 구성하기 위해서는 각 컴퓨터에 두 개의 네트워크 카드가 장착되어야 하며 각 카드마다 별도의 IP 주소가 할당되고, 특히 클러스터 서버를 구성하는 두 대의 서버에는 별도의 클러스터 네트워크를 위하여 세 개의 네트워크 카드가 장착된다. 각 컴퓨터에서의 응용 프로그램들은 다른 컴퓨터 내의 응용 프로그램과 통신을 하고자 할 때 메인 네트워크의 상태를 체크하여 정상 동작할 경우는 메인 네트워크를 사용하고 이상이 있을 경우는 백업 네트워크를 사용

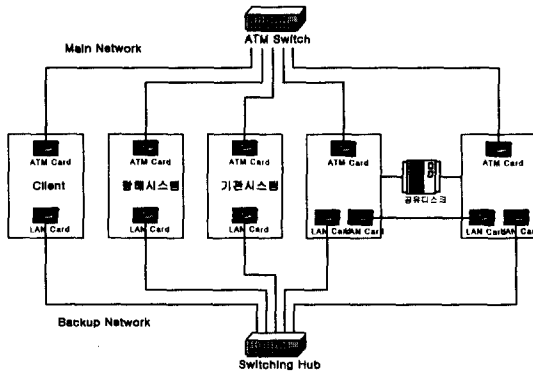


그림 11. 시험용 LAN 구성

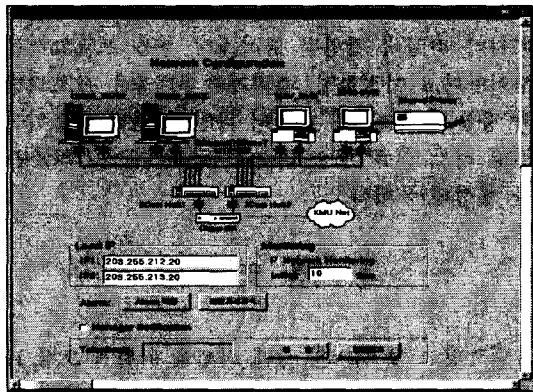


그림 12. 네트워크 모니터링 시스템

하게 된다. 이러한 동작은 각 컴퓨터 내의 네트워크 모니터링 및 스위칭 프로그램에 의해 수행되며, 그림 12는 네트워크 모니터링 시스템의 실행 화면을 나타낸다¹¹⁾.

4.1 네트워크 고장허용 시험

네트워크 고장허용 시험을 위하여 시험용 LAN에서의 메인 네트워크에 강제 장애를 발생시키고 결과를 시험하였다. 시험용 LAN에서 메인 네트워크는 203.255.212 주소를 할당하였고 백업 네트워크는 203.255.213 주소를 할당하였다. 메인 네트워크에 장애가 발생하였을 경우 네트워크 모니터링 시스템은 알람을 통하여 경고 메시지를 보내주었으며, "HOSTS" 파일의 내용은 그림 13과 같이 각

호스트 이름에 대한 IP 주소가 메인 주소에서 백업 주소로 스위칭 되어 백업 네트워크를 통하여 접속됨을 확인할 수 있었다.

# This is a sample HOSTS file		# This is a sample HOSTS file	
127.0.0.1	localhost	127.0.0.1	localhost
203.255.212.20	osdb_serv	203.255.213.20	osdb_serv
203.255.212.21	osdb_back	203.255.213.21	osdb_back
203.255.212.22	osdb_clus	203.255.213.22	osdb_clus
203.255.212.23	nav_serv	203.255.213.23	nav_serv
203.255.212.24	eng_serv	203.255.213.24	eng_serv

그림 13. "HOSTS" 파일의 스위칭

4.2 서버 고장허용 시험

선박 데이터베이스 서버는 두 대의 시스템을 하나의 클러스터 서버로 구성하였는데, 각 시스템의 호스트 이름은 osdb_serv, osdb_back으로, 클러스터 이름은 osdb_clus로 지정하였다. 따라서 각 클라이언트들은 osdb_clus란 호스트 이름으로 데이터베이스 서버에 액세스하게 되며, 어느 서버 시스템이

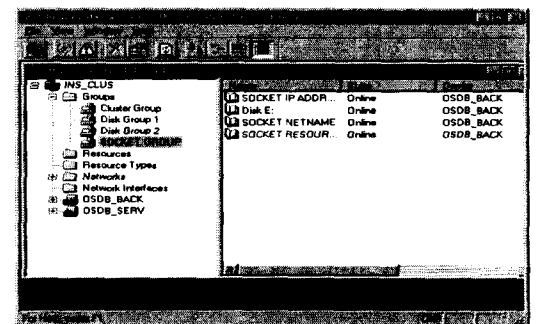
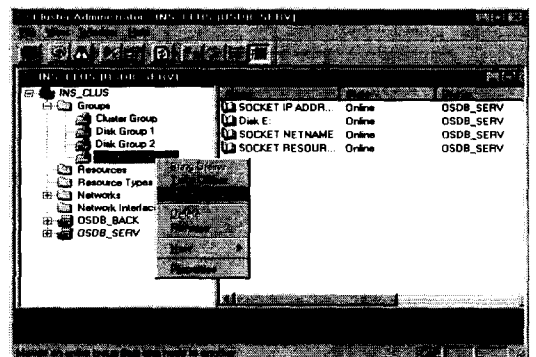


그림 14. 클러스터 작업인계

서비스를 제공하는 지는 알 수 없다. 서버의 고장허용 시험을 위해서는 클러스터 관리자 프로그램을 이용하여 osdb_serv 내의 소켓 그룹을 강제적으로 osdb_back 서버로 이동시켰으며, 그림 14와 같이 작업인계가 이루어짐을 확인할 수 있었다.

5. 결론

현재 선박 자동화 시스템 기술의 발달로 선박 내에서도 네트워크는 물론 다양한 데이터의 효율적인 관리를 위한 데이터베이스 구축이 늘어나고 있는 추세이며, 선박의 현재 상태나 제어정보는 항해 중인 선박의 안전과 더불어 육상과의 정보교환에도 중요한 위치를 차지하고 있다. 이로 인해 선박 내 네트워크와 여러 정보를 관리하는 데이터베이스는 항상 동작 상태에 있어야 한다. 따라서 이에 대응하는 네트워크 장애 대책과 서버의 가용성을 높이는 문제가 대두된다.

본 논문에서는 선박의 안전성과 신뢰성을 위한 선박 LAN 구축을 위하여 네트워크, 서버 및 디스크의 고장허용 기법을 고찰하고, 이를 토대로 단일 네트워크의 문제점을 해결하기 위한 이중 네트워크의 구성, 데이터베이스 및 주요 서버 프로그램을 관리하는 서버의 장애를 해결한 클러스터 서버 구성 등 신뢰성 있는 선박 LAN 구축 방안을 제안하였다. 또한 제안한 방법대로 실제 테스트 LAN을 구성하고 네트워크 및 서버의 장애 시험을 통하여 신뢰성을 확인하였다.

향후 연구로는 클러스터가 지원하는 노드의 수를 늘리고 확장성이 개선되어야 할 것이며, 선박 내 서버에 이상이 발생했을 경우 빠른 서비스를 제공하기 위해서는 작업인계 시간이 향상되어야 할 것이다. 또한 선박 내 네트워크 장애에 대한 가용성과 확장성을 확보하기 위해서는 뛰어난 관리기능이 필요하다. 시스템이 단순히 네트워크로 묶이는 것이 아니라 상호 감시 하에 연결되는 복잡한 구조인 만큼 관리 영역이 보다 커지는 것이다. 에러 유형에 따라 시스템을 재가동하는 것이 시간적으로 빠를 수 있으며, 자원활용 측면에서도 더 유리할 수 있

다. 따라서 에러의 사전 감지, NMS(Network Management System)와 같은 선박 내 항해, 기관 시스템 등의 다양한 모니터링 기능에 대한 연구가 진행되어야 할 것이다.

참고문헌

- 1) <http://www.digital.co.kr>, "Windows NT용 디지털 클러스터 백서," 디지털 매거진, 1996.
- 2) Elizabeth Clark, "Server Clustering: The Good of the Many," Network, Magazine, Aug., 1997.
- 3) Behrouz Forouzan, Introduction to Data Communication and Networking, McGraw-Hill, 1998.
- 4) McDermott WR, Tri JL, Mitchell MP, Levens SP, Wondrow MA, Huie LM, Khandheria BK, Gilbert BK, "Optimization of wide-area ATM and local-area ethernet/FDDI network configurations for high-speed telemedicine communications employing NASA's ACTS," IEEE Network, Vol. 13, No. 4, pp.30-38, 1999.
- 5) Rob Short, Rod Gauche, John Vert and Mike Massa, Windows NT Clusters for Availability and Scalability, Microsoft Research White Paper, 1998.
- 6) Lori Merric, Clustering Support for Microsoft SQL Server, Microsoft Research White Paper, 1997.
- 7) Thomasian A, Menon J, "RAID 5 Performance with Distributed Sparring," IEEE Transaction on Parallel & Distributed Systems, Vol. 8, No.6, pp.640-657, 1997.
- 8) Werner Vogels, "The Design and Architecture of the Microsoft Cluster Service," Proceedings of FTCS'98, Munich, Germany, June, 1998.
- 9) Vogels W, Dumitriu D, Agrawal A, Chia T, and Guo K, "Scalability of the Microsoft Cluster Service," Proceedings of the Second

10 韓國航海學會誌 第24卷 第1號, 2000

Usenix Windows NT Symposium, Seattle, WA, August 1998.

10) 김 영 수, 조 익 성, 임 재 홍, "클러스터 인식 응용 프로그램의 구현 기법," 한국해양정보통신학회 '99 추계학술대회 논문집, 제 3 권, 제 2 호, pp.252-259, 1999년 11월.

11) 최 소 영, 단순 네트워크 관리 시스템의 설계 및 구현에 관한 연구, 한국해양대학교 대학원 공학석사 학위논문, 1999년 8월.