



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

이더넷 기반 선박 통합 네트워크를 위한 미들웨어의 설계 및 구현

A Design and Implementation of Middleware
for Ethernet based Shipboard Integration Networks



지도교수 이장세

2011년 2월

한국해양대학교 대학원

컴퓨터공학과

황 훈 규

이 논문을 황훈규의 공학석사 학위논문으로 인준함.

위원장 이 서 정



위 원 박 휴 찬



위 원 이 장 세



2010년 12월 24일

한국해양대학교 대학원

목 차

제 1 장 서론	1
제 2 장 관련 연구	3
2.1 e-Navigation 환경의 선박 네트워크	3
2.1.1 e-Navigation의 정의 및 목적	3
2.1.2 e-Navigation 환경의 선박 네트워크 구조	5
2.2 선박 통합 네트워크 표준	7
2.2.1 IEC 61162-4 시리즈	7
2.2.2 IEC 61162-450	11
2.3 미들웨어	12
제 3 장 IEC 61162-450 분석	13
3.1 네트워크 기능(NF : Network Function block)	13
3.2 시스템 기능(SF : System Function block)	14
3.2.1 TAG/Sentence	15
3.2.2 바이너리 이미지	20
3.2.3 오류 로그	22
제 4 장 선박 통합 네트워크 미들웨어 설계 및 구현	23
4.1 선박 통합 네트워크 구조 정의	23
4.2 선박 통합 네트워크 미들웨어 설계	25
4.2.1 네트워크 기능 모듈	26
4.2.2 시스템 기능 모듈	27
4.2.3 서비스 모듈	31
4.3 선박 통합 네트워크 미들웨어 구현	33
4.3.1 데이터 형식 정의	33
4.3.2 클래스 구현	34

제 5 장 선박 통합 네트워크 미들웨어 시험	37
5.1 시험 요구사항	37
5.2 선박 통합 네트워크 미들웨어	38
5.2.1 TAG/Sentence 송수신	38
5.2.2 바이너리 이미지 송수신	41
5.2.3 오류 로그	43
5.3 선박 통합 네트워크	45
제 6 장 결론 및 향후 연구	46
참고문헌	47



그림 목차

그림 2.1 e-Navigation 환경의 선박 네트워크	5
그림 2.2 IEC 61162-4 시리즈 구조	8
그림 2.3 IEC 61162-450 기반 선박 통합 네트워크	10
그림 2.4 IEC 61162-450 구조	11
그림 3.1 IEC 61162-450 데이터그램 구조	14
그림 3.2 TAG/Sentence 메시지 구성의 예	16
그림 3.3 DPT Sentence Formatter 구성	16
그림 3.4 'g' 인자 코드 구성 예	18
그림 3.5 's', 'd' 인자 코드 구성 예	19
그림 3.6 'n' 인자 코드 구성 예	19
그림 3.7 바이너리 이미지 메시지 내부 구조	21
그림 3.8 내부 로그 메시지 구조	22
그림 3.9 외부 로그 메시지 구조	22
그림 4.1 IEC 61162-450 기반 선박 통합 네트워크	24
그림 4.2 각 모듈의 동작 흐름도(송신)	25
그림 4.3 각 모듈의 동작 흐름도(수신)	26
그림 4.4 TAG/Sentence 메시지의 유효성 검사 알고리즘	28
그림 4.5 Sentence 버퍼 구조	29
그림 4.6 바이너리 이미지의 송신 알고리즘	30
그림 4.7 바이너리 이미지의 수신 알고리즘	31
그림 4.8 TAG 및 Sentence Formatter 구조체	34
그림 4.9 선박 통합 네트워크 미들웨어의 클래스 다이어그램	35
그림 4.10 API 함수 호출시 내부 동작의 예	36

그림 5.1 TAG/Sentence 메시지 송신 시험	38
그림 5.2 TAG/Sentence 메시지 수신 시험	39
그림 5.3 올바르지 않은 Sentence 메시지 송신 시험	40
그림 5.4 올바르지 않은 Sentence 메시지 수신 시험	40
그림 5.5 바이너리 이미지 송신 시험	41
그림 5.6 바이너리 이미지 수신 시험	42
그림 5.7 내부 로그 저장 결과	43
그림 5.8 외부 로그 저장 결과	44
그림 5.9 시험을 위해 구축된 선박 통합 네트워크.....	45



표 목차

표 2.1 e-Navigation의 목적	4
표 2.2 미들웨어의 기능	12
표 3.1 IEC 61162-450 메시지 형식	13
표 3.2 IEC 61162-450 메시지 헤더	14
표 3.3 TAG/Sentence 메시지 구조	15
표 3.4 TAG 블록 인자 코드	17
표 3.5 바이너리 이미지 메시지 구조	20
표 4.1 서비스 모듈 API 함수	32
표 4.2 Sentence Formatter 및 의미	33



A Design and Implementation of Middleware for Ethernet based Shipboard Integration Networks

Hun-Gyu Hwang

*Department of Computer Engineering,
Graduate School of
Korea Maritime University*

Abstract

Shipboard integration networks, that are a part of e-Navigation, provide interconnection and integrated management for information from shipboard devices. The IEC 61162-4 series standard has established for the shipboard integration networks. But the standard has too many requirements and is difficult to be implemented. So the standard has been decided to change IEC 61162-450.

This thesis proposes the middleware for ethernet based shipboard integration networks. To do this, first, this thesis compares IEC 61162-4 series with IEC 61162-450 and analyzes requirements of IEC 61162-450. Second, this thesis designs and implements a Network Function module and a System Function module based on the analysis of IEC 61162-450. The Network Function module sends and receives datagram using UDP multicast. The System Function module processes TAG/Sentence and binary messages, checks validity of the messages, and records error logs. Third, this thesis designs and implements a service module for interfacing the Network Function module and the System Function module to applications, and develops the middleware to integrate the Network Function module, the System Function module, and the service module. Finally, this thesis tests the middleware for effectiveness.

제 1 장 서 론

e-Navigation 환경의 선박 네트워크의 구현을 위한 요소 중 하나인 선박 통합 네트워크(Shipboard Integration Networks)는 선박에 탑재된 여러 장비에서 발생하는 다양한 정보를 상호 교환하고 통합 관리하기 위한 목적을 가지는 네트워크이다[1].

IEC(International Electro-technical Commission)는 2001년에 선박 통합 네트워크의 표준인 IEC 61162-4 시리즈를 발표하였지만 이는 요구사항이 많고 시스템 구조가 복잡하여 구현의 어려움 등 여러 문제점이 있었다. 이러한 여러 문제점 때문에 IEC TC(Technical Committee) 80 총회에서는 2009년에 더 이상 IEC 61162-4 시리즈를 선박 통합 네트워크의 표준으로 유지하지 않기로 결정하였으며 이를 대체할 표준인 IEC 61162-450을 제정하는 중에 있다[2-4].

IEC 61162-450은 기존 IEC 61162-4 시리즈에 비해 시스템 구조나 요구사항 등이 단순해진 것이 특징이다. IEC 61162-450을 기반으로 하는 선박 통합 네트워크에서 여러 시스템 간의 통신을 위해서는 네트워크 기능(NF : Network Function block)과 시스템 기능(SF : System Function block)이 필요하다. NF는 선박 통합 네트워크 내에 있는 시스템 간의 상호 연결 및 데이터그램 송수신 등의 기능을 하며 SF는 송수신된 데이터그램의 메시지 처리, 유효성 검사, 오류 로그 등의 기능을 한다[3].

이 논문에서는 IEC 61162-450 기반의 선박 통합 네트워크를 위한 미들웨어 설계 및 구현에 관한 내용을 다룬다. 미들웨어의 설계 및 구현을 위해, 먼저 이 논문의 연구 배경이 되는 e-Navigation 환경의 선박 네트워크 및 그 구성 요소에 관한 내용을 기술한다. 특히 선박 통합 네트워크에 관한 내용을 자세히 기술하며 선박 통합 네트워크의 기존 표준인 IEC 61162-4 시리즈와 제정 중인 IEC 61162-450 표준을 비교하고 분석한다. 또한 IEC 61162-450 기반의 선박 통합 네트워크에서 여러 시스템 간의 정보 교환 및 처리를 위해 사용되는 요소인

NF 모듈 및 SF 모듈을 개발하고 이에 추가적으로 선박 통합 네트워크의 여러 시스템이 NF 모듈과 SF 모듈의 기능을 이용할 수 있도록 인터페이스를 제공하는 서비스 모듈을 개발한다. 끝으로 NF 모듈, SF 모듈, 서비스 모듈을 통합한 미들웨어를 개발하고 개발된 미들웨어의 유용성을 시험한다.

이 논문의 구성은 다음과 같다. 먼저 2장에서 선박 통합 네트워크 미들웨어 개발의 배경이 되는 관련 연구를 기술하고 3장에서 선박 통합 네트워크의 표준인 IEC 61162-450을 분석한다. 또한 4장에서 분석된 내용을 바탕으로 선박 통합 네트워크 미들웨어의 개발에 관하여 다루고 5장에서 구현된 미들웨어의 유용성을 시험하며 6장의 결론 및 향후 연구로 끝을 맺는다.



제 2 장 관련 연구

2.1 e-Navigation 환경의 선박 네트워크

2.1.1 e-Navigation의 정의 및 목적

e-Navigation은 전자적인 방법으로 해상 안전·보안과 해양 환경 보호를 위해 출발지에서 목적지까지 항해와 관련된 여러 서비스를 개선하여 선박과 육상의 해양 정보 수집, 통합, 교환, 표현, 분석이 잘 조화될 수 있도록 하는 것이다[5].

e-Navigation은 선박의 전 항해 구간에 대해 안전 항해 및 효율적인 해상 운송 서비스를 위한 정보의 중요성과 정보 이용의 효율 극대화를 위한 수단의 필요성을 강조하고 있다. 또한 선박에서 육상까지 해상 정보의 수집에서부터 활용에 이르기까지 효율성의 극대화를 위하여 표 2.1과 같은 11가지 목적을 가지는데 이 11가지 항목 중 3, 7, 8 등이 e-Navigation 환경의 선박 네트워크의 구성과 밀접한 관련이 있다[6]. 다음 항에서 선박 네트워크의 구조 및 구성 요소에 대해서 기술한다.

표 2.1 e-Navigation의 목적

Table 2.1 Purposes of the e-Navigation

항목	내 용
1	수로, 기상, 항해 정보와 항해 위험을 고려하여 안전하고 보안이 확보된 선박의 항해를 지원한다.
2	육상/연안 관제시설에서 해상교통량의 관측 및 관리를 수월하게 한다.
3	선박에서 선박으로, 선박에서 육상으로, 육상에서 선박으로, 육상에서 육상 및 이용자 간의 데이터 교환을 포함한 통신을 수월하게 한다.
4	해상운송과 물류의 효율을 개선할 수 있는 기회를 제공한다.
5	해난 사고 대응, 수색 및 구조 작업이 효율적으로 운영될 수 있도록 지원한다.
6	안전 시스템(safety-critical system)에 요구되는 적절한 정확도, 무결성, 연속성 등에 대한 정량적 수준의 예를 제시한다.
7	항해 안전은 극대화하고 이용자들 간의 잘못된 상황 해석이나 혼동으로 인하여 발생할 수 있는 위험을 최소화시킬 수 있도록 선박과 육상에 여러 항해 정보를 통합 및 가공하여 제공한다.
8	통합 및 가공된 항해 정보를 선박과 육상에 제공하여 이용자의 작업 부하를 줄이고 이용자가 보다 많은 시간을 의사 결정에 할애할 수 있도록 지원한다.
9	항해 장비 개발 및 구현 과정에서 이용자가 장비에 익숙해지기 위한 훈련과 관련하여 적응 요구사항을 반영한다.
10	글로벌 커버리지, 일관된 표준 및 배치, 장비들 사이의 시스템, 심벌, 운영 절차와 관련하여 상호호환성, 상호운영성을 갖도록 하여 이용자들 간에 일어날 수 있는 혼동을 방지할 수 있다.
11	모든 잠재적인 해상 이용자가 이용할 수 있도록 시스템은 다양한 크기의 선박 및 육상시설에 설치 가능하도록 비례 축소되어야 한다.

2.1.2 e-Navigation 환경의 선박 네트워크 구조

그림 2.1은 e-Navigation 환경에서의 선박 네트워크 구성을 나타낸 것이다. 선박 네트워크는 특정한 목적을 갖는 여러 장비가 연결된 전용 연결 네트워크(Dedicated Connections)를 제외하고 선박의 각종 장비 사이의 정보 교환을 위한 장비 네트워크(Instrument Networks), 선박 내에 있는 시스템의 감시·제어·통합 정보 관리를 위한 선박 통합 네트워크(Shipboard Integration Networks), 4S(Ship to Ship, Ship to Shore) 멀티미디어 통신을 포함하는 관리 네트워크(Administrative Networks)로 구성된다[7,8].

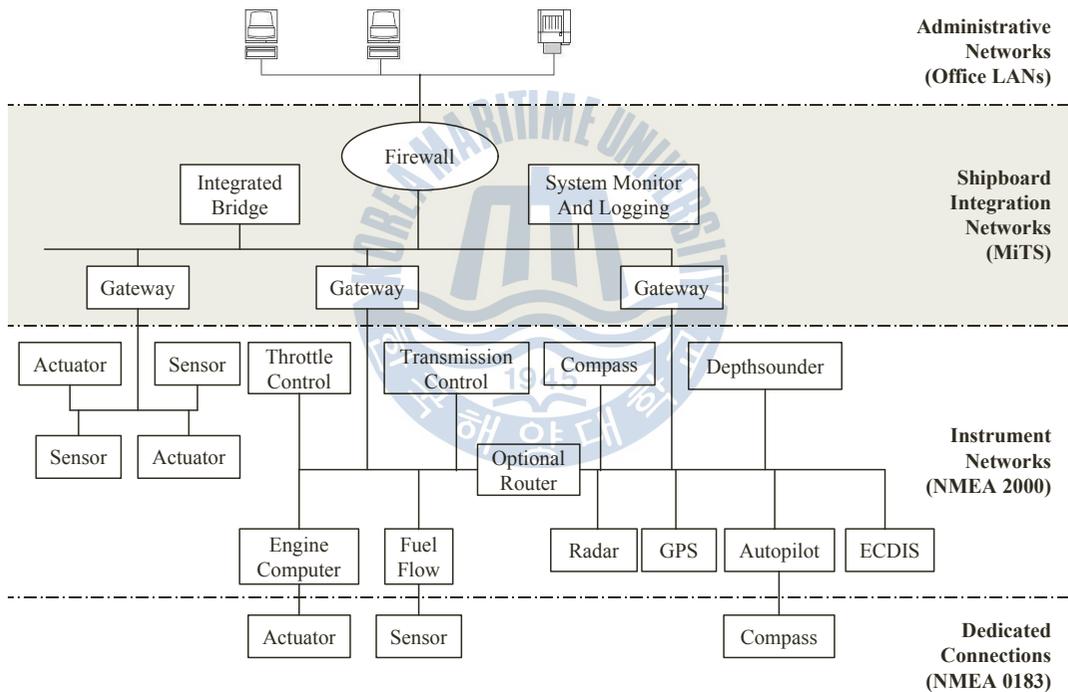


그림 2.1 e-Navigation 환경의 선박 네트워크

Fig. 2.1 Shipboard networks of the e-Navigation environment

e-Navigation 환경의 선박 네트워크를 구성하는 각 네트워크의 기능 및 특징은 다음과 같다.

(1) 장비 네트워크

장비 네트워크는 선박 내의 여러 장비가 실시간으로 정보를 교환할 수 있도록 해주는 역할을 하며 NMEA 2000(IEC 61162-3) 표준을 통해 다양한 선박 네트워크 기술이 혼재하는 것을 방지하고 선박 내 각 장비 간의 호환성을 높이기 위한 목적으로 승인 및 채택되었다[9]. 장비 네트워크는 CAN(Controller Area Network)을 인터페이스로 채택하고 있으며 CAN은 거리에 따라 통신 속도가 다르게 정해진다. 장비 네트워크의 경우, 최대 200미터의 거리에서 250kbps의 속도가 보장된다. 또한 CAN은 최대 50개의 물리 노드(node)와 252개의 논리 노드를 가진다[10].

(2) 선박 통합 네트워크

선박 통합 네트워크는 선박 내의 여러 장비에서 발생하는 여러 정보의 통합 관리 및 교환을 목적으로 하는 네트워크이다. 2001년, 선박 통합 네트워크의 표준으로 IEC 61162-4 시리즈가 채택되었으나 이는 여러 문제점을 가지고 있었기 때문에 선박 통합 네트워크의 표준을 IEC 61162-450으로 대체하기로 결정하였다. IEC 61162-450은 현재 제정 작업이 진행되고 있으며 2011년 8월에 정식 버전이 발표될 예정이다[4].

NMEA(National Marine Electronics Association) 2000 표준을 따르는 장비 네트워크 내의 여러 장비로부터 수집된 정보는 게이트웨이(gateway)를 통해 선박 통합 네트워크의 형식으로 변환되어 들어오게 된다. 그 뒤, 선박 통합 네트워크 내에 있는 시스템의 여러 어플리케이션(application)은 각각의 기능에 맞는 동작을 위해 선박 통합 네트워크 내에 들어온 정보를 이용하게 된다[11,12]. 선박 통합 네트워크 및 그 표준에 대한 자세한 내용은 다음 절에서 기술한다.

(3) 관리 네트워크

관리 네트워크에 포함되어 있는 4S는 선박과 선박 간, 선박과 육상 간의 통신을 위하여 선박에 탑재되어 있는 여러 통신 장비와 육상의 다양한 통신 인프라를 통합하는 역할을 한다. 4S는 정오 보고(noon report) 등을 위하여 선박과 육상이 일시적으로 연결되는 것이 아니라 특정 시간 마다 이-메일 등의 다양한 서비스를 제공하기 위해 가장 효율적인 통신 매체를 선택적으로 이용하여 육상으로 연결하는 것이다. 즉, 육지와 가까운 연안에서는 CDMA, WiBRO 등 고속 통신이 가능한 통신 매체를 이용하고 근해에서는 HF 통신을 이용하며 원양에서는 위성통신을 이용한다[8].

2.2 선박 통합 네트워크 표준

선박 통합 네트워크에서 사용되는 표준인 MiTS(Maritime information Technology Standard)는 선박에서 발생하는 다양한 정보의 통합 관리 및 상호 교환을 위하여 제안되었다. 2001년 IEC는 선박 통합 네트워크에서 여러 시스템의 상호 연결 및 정보 교환을 위한 표준인 IEC 61162-4 시리즈를 발표하였지만 이는 요구사항이 복잡하고 많으며 구현의 어려움 등 여러 문제점을 가지고 있었다. 그 때문에 2009년 IEC TC 80 총회에서는 더 이상 61162-4 시리즈를 표준으로 유지하지 않기로 결정하였으며 기존 표준에 비해 요구사항이나 시스템 구조가 경량화된 표준인 IEC 61162-450을 제정하는 중에 있다[3,4].

2.2.1 IEC 61162-4 시리즈

MiTS는 이더넷 기반 선박 통합 네트워크의 표준으로 유럽연합(EU)의 PISCES(Protocols for Integrated Ship Control and Evaluation of Situations) 프로젝트로부터 시작되었으며 이를 기반으로 IEC 61162-4 시리즈가 채택 및 발표되었다[13].

IEC 61162-4 시리즈의 구조는 그림 2.2와 같으며 T-프로파일(Transport-Profile)과 A-프로파일(Application-Profile) 등으로 구성된다. IEC 61162-4 시리즈를 기반으로 하는 선박 통합 네트워크 내의 시스템은 상호 간의 정보 교환을 위하여 TCP/IP (Transmission Control Protocol/Internet Protocol) 기반의 통신을 한다.

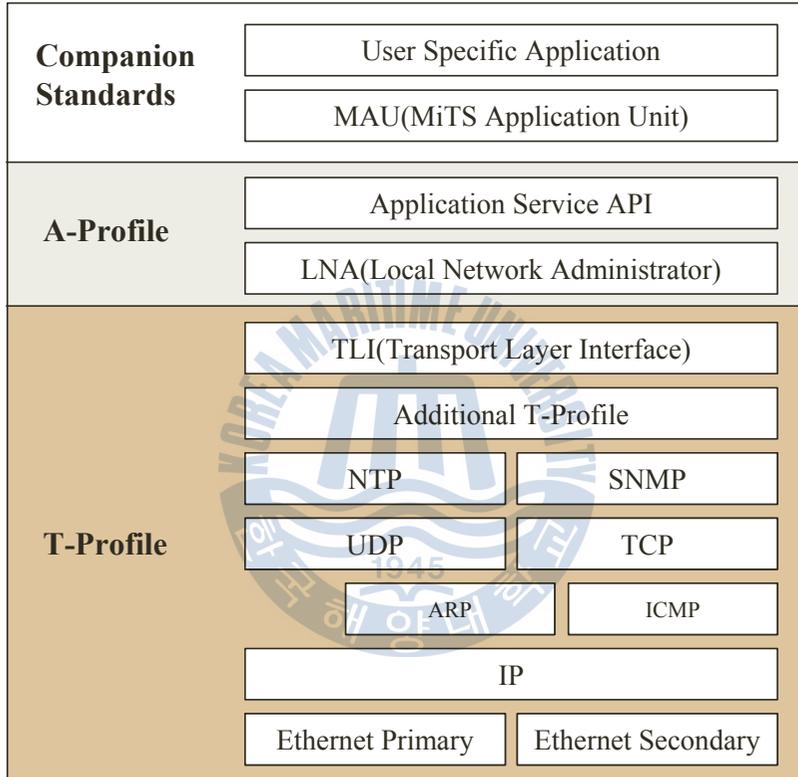


그림 2.2 IEC 61162-4 시리즈 구조
 Fig. 2.2 Architecture of the IEC 61162-4 series

T-프로파일은 OSI(Open Systems Interconnection) 7계층 중 1계층부터 4계층까지를 포함하며 네트워크 연결과 전송을 담당하는 부분이다. T-프로파일은 네트워크 주소관리 서비스, 신뢰적 메시징 서비스, 신뢰적 스트림 서비스, 비-신뢰적 데이터그램 서비스, 시스템 관리 서비스, 시간 분배 서비스, 예외처리 및 보고 서비스 등을 제공한다.

A-프로파일은 OSI 7계층 중 5계층부터 7계층까지를 포함하며 통신 시스템의 프로토콜을 담당하는 부분이다. A-프로파일은 LNA(Local Network Administrator)와 협력 표준(Companion Standard)의 MAU(MiTS Application Unit) 간의 서비스 정의와 구현을 제공하며 자동 설정, 데이터 객체를 이용한 클라이언트-서버 구조, 연결 지향적, 트랜잭션 지향적, 신뢰적 전송 등의 특성을 가진다[2]. 또한 A-프로파일이 T-프로파일 내부 서비스에 접근하기 위해 필요한 인터페이스를 정의하는 TLI(Transport Layer Interface)는 장비에 의존적이다. 앞서 설명한 기능 이외에도 IEC 61162-4 시리즈는 요구하는 기능이 많기 때문에 시스템 구조가 크고 복잡하다.

[11-13]에서 제안한 IEC 61162-4 시리즈 기반의 선박 통합 네트워크 구조는 그림 2.3과 같이 게이트웨이, 미들웨어 서버, 모니터링 시스템으로 구성되어 있다. 게이트웨이는 여러 선박 장비로부터 수집된 정보를 가공하여 선박 통합 네트워크의 형식으로 변환해주는 기능을 한다. 미들웨어 서버(middleware server)는 그 정보를 처리하여 데이터베이스에 저장하는 기능을 하고 모니터링 시스템은 데이터베이스에 저장된 정보를 사용자에게 표시하는 기능을 한다. 각 시스템은 선박 통합 네트워크에 연결되어 있지만, 실제 정보는 게이트웨이에서 미들웨어 서버로, 미들웨어 서버에서 모니터링 시스템으로 흐르는 계층적인 특성을 가진다[13].

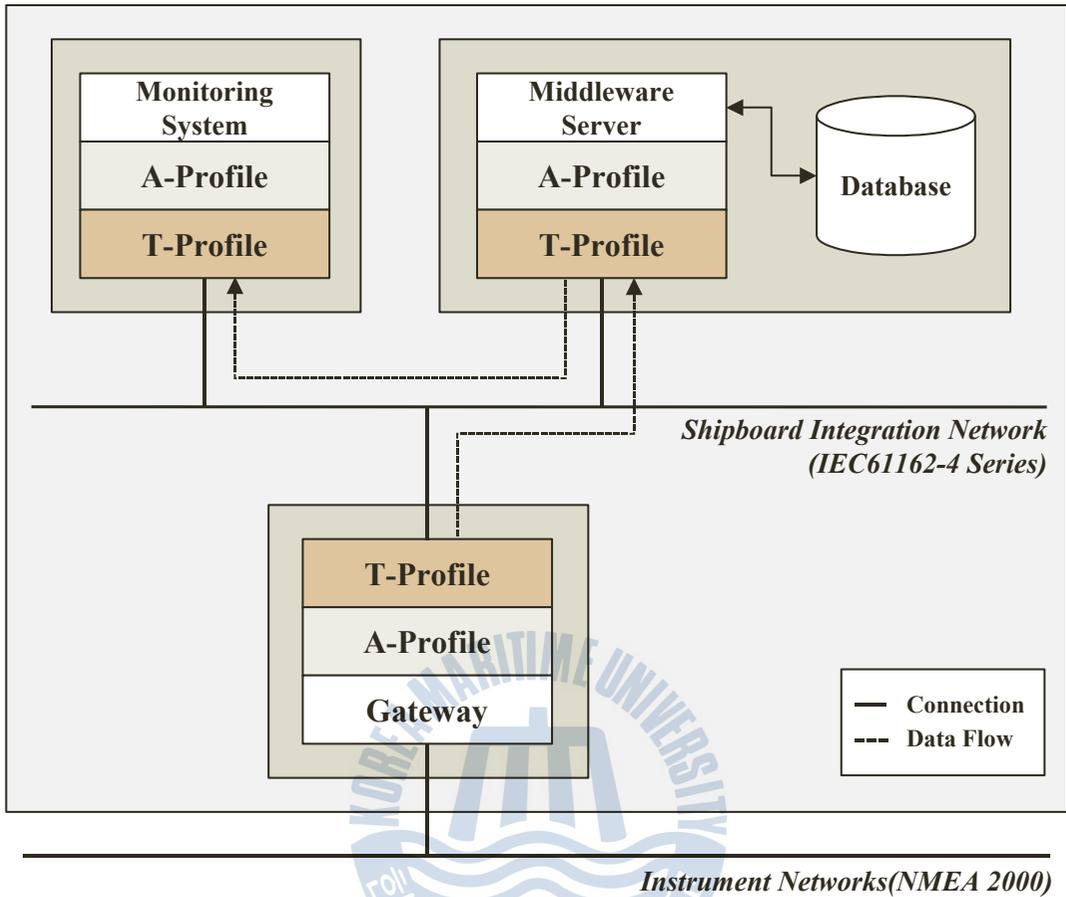


그림 2.3 IEC 61162-4 시리즈 기반 선박 통합 네트워크

Fig. 2.3 Shipboard integration network based on the IEC 61162-4 series

2.2.2 IEC 61162-450

IEC 61162-450의 구조는 그림 2.4와 같으며 그림 2.2의 IEC 61162-4 시리즈의 구조에 비해 비교적 단순해졌다는 것을 확인할 수 있다. 네트워크 기능(NF : Network Function block)과 시스템 기능(SF : System Function block)이 IEC 61162-450의 핵심 요소이다. IEC 61162-450은 TCP/IP를 이용하는 기존의 IEC 61162-4 시리즈와는 달리 UDP/IP를 이용하며 멀티캐스트(multicast) 방식으로 통신한다.

Tag를 기반으로 하는 메시지 형식을 사용하여 정보를 교환했던 IEC 61162-4 시리즈와는 달리 IEC 61162-450은 일반적인 정보의 교환에는 NMEA 0183 버전 4 기반의 TAG/Sentence 메시지 형식을 사용하며 바이너리 이미지의 교환에는 IEC 61162-450에서 독자적으로 정의하고 있는 형식을 사용한다. 또한 향후 IEC 61162-3(NMEA 2000)도 지원할 예정이다[2-4].

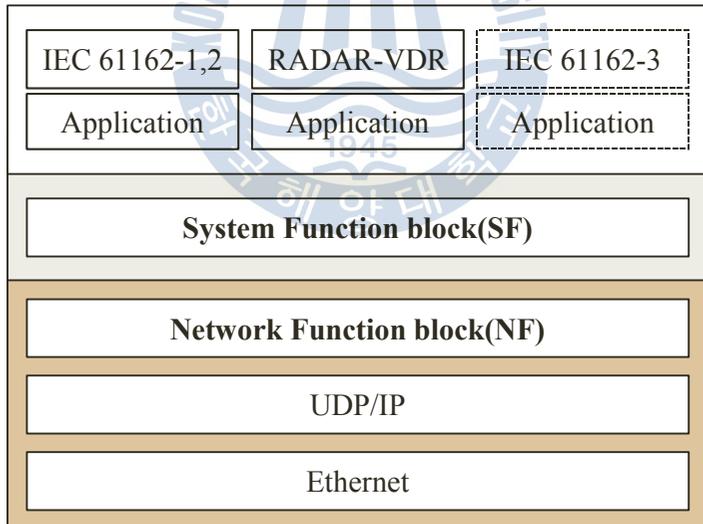


그림 2.4 IEC 61162-450 구조

Fig. 2.4 Architecture of the IEC 61162-450

2.3 미들웨어

미들웨어(middleware)라는 개념은 1980년대 후반에 네트워크 연결 관리 소프트웨어를 설명하기 위해 처음 등장하였으며 네트워크 기술이 성장하는 1990년대 중반부터 개념이 널리 퍼지기 시작하였다. 1990년대 초반에는 주로 데이터베이스 분야와 연관되었으나 이후 분산 응용 서비스 분야로 점차 확대되었다.

미들웨어는 분산되어 있는 여러 시스템을 상호 연결하여 업무를 처리하는 소프트웨어이다. 미들웨어는 시스템의 유형, 데이터베이스의 종류, 네트워크의 형태, 위치에 관계없이 작업을 처리할 수 있도록 해주는 기능을 한다. 미들웨어는 아래 표 2.2와 같은 주요 기능을 가지며[14] 이 논문에서 다루는 선박 통합 네트워크 미들웨어는 이 항목 중 3, 4, 7, 10 등의 기능을 제공한다.

표 2.2 미들웨어의 기능

Table 2.2 Functions of the middleware

항목	기능
1	프로세스 관리
2	분산 트랜잭션 관리
3	부하 조절
4	장애 대책
5	보안 기능
6	네이밍(naming) 서비스
7	편리한 API 및 다양한 통신 방식 지원
8	시스템 관리
9	멀티도메인 및 다양한 게이트웨이 모듈 제공
10	신뢰성을 보장하는 큐(reliable queue) 제공

제 3 장 IEC 61162-450 분석

3.1 네트워크 기능(NF : Network Function block)

NF는 선박 통합 네트워크 내 시스템 간의 연결을 담당하며 인터페이스로 이더넷을 채택하고 있다. NF는 UDP 멀티캐스트 통신을 하며 멀티캐스트를 위한 IP 주소는 IP 버전 4 기반의 239.192.0.1부터 239.192.0.64까지의 범위를 사용한다. 또한 표 3.1에서 볼 수 있는 메시지 형식에 따라 49152번부터 65535번 사이의 포트를 사용하며 송수신시 각 데이터그램의 최대 크기는 1472 바이트이다[3].

표 3.1 IEC 61162-450 메시지 형식
Table 3.1 Message types of the IEC 61162-450

형식	설명
MISC	일반적인 데이터
TGTD	AIS 타겟(target) 및 추적 데이터
SATD	높은 업데이트율을 가진 데이터
NAVD	항해 데이터
VDRD	VDR 데이터
RCOM	무선 통신 장비
TIME	시간 전송 장비
PROP	소유자 및 사용자 명세 SF
USR1~USR8	사용자 정의 전송 그룹
바이너리 이미지 데이터	
재전송 가능한 바이너리 이미지 데이터	

IEC 61162-450 기반의 선박 통합 네트워크에서 시스템 간의 정보 교환을 위해 사용되는 데이터그램의 구조는 그림 3.1과 같이 14 바이트의 프레임(frame) 헤더, 20 바이트의 IP 헤더, 8 바이트의 UDP 헤더, 0~1472 바이트의 IEC 61162-450 메시지, 4 바이트의 CRC(Cyclic Redundancy Check)로 구성되어 있다. IEC 61162-450 메시지를 구성하고 처리하는 역할은 SF가 담당하고 있으며 다음 절에서 자세히 기술한다.

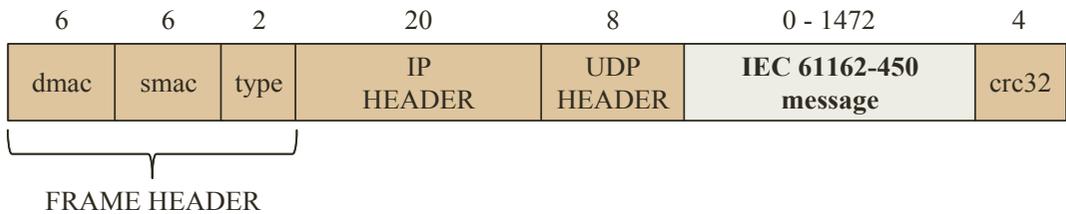


그림 3.1 IEC 61162-450 데이터그램 구조
Fig. 3.1 Datagram structure of the IEC 61162-450

3.2 시스템 기능(SF : System Function block)

이때, 크게 두 가지 형식의 메시지를 구별하여 처리하는데, NF를 통해 송수신하는 메시지가 TAG/Sentence 형식인지 바이너리 이미지 형식인지를 구별하기 위해 각 메시지의 앞에는 표 3.2와 같은 헤더(header)를 추가한다[3].

표 3.2 IEC 61162-450 메시지 헤더

Table 3.2 Message headers of the IEC 61162-450

종류	설명
UdPbC	NMEA 0183 버전 4 형식의 TAG/Sentence 메시지
RaUDP	바이너리 이미지 메시지
RrUDP	재전송 가능한 바이너리 이미지 메시지

3.2.1 TAG/Sentence

IEC 61162-450에서 TAG/Sentence 메시지의 송수신 형식은 NMEA 0183 버전 4를 기반으로 정의하고 있으며 바이너리 이미지 메시지의 송수신 형식은 IEC 61162-450에서 정의하고 있다. TAG/Sentence 메시지는 일반적으로 표 3.3과 같은 형식으로 구성된다. '\$'는 일반적인 Sentence인 경우의 시작을 표시해주고 '!'는 캡슐화된 Sentence인 경우의 시작을 표시한다. 장비를 식별할 수 있도록 해주는 장비 식별자(talker identifier)는 일반적으로 2 바이트의 문자로 구성되어 있으며 Sentence의 형식을 알려주는 Sentence Formatter는 3 바이트의 문자로 구성되어 있다. 그 뒤, 실제 정보를 담고 있는 데이터 필드가 위치하게 되는데 이 데이터 필드는 ','로 구분한다. 마지막 데이터 필드의 뒤에는 '*' 및 Sentence의 체크섬 확인을 위한 두 자리 16진수가 위치하며 <CR><LF>로 Sentence가 끝난다[15].

표 3.3 TAG/Sentence 메시지 구조

Table 3.3 Structure for the TAG/Sentence message

항목	설명
"UdPbC"	메시지 헤더
null	메시지 헤더 구분자
\ TAG block \	TAG 블록(선택 사항)
'\$' or '!'	Sentence의 시작
char[2]	장비 식별자(일반적인 경우)
char[3]	Sentence Formatter
[' , '<data>]	데이터 필드
'*'	체크섬(checksum) 필드 구분자
hh	체크섬 필드 (2자리의 16진수)
<CR><LF>	Sentence의 끝

TAG/Sentence 메시지에서 TAG는 최대 80 바이트, Sentence는 최대 82 바이트의 크기를 가진다. 또한 TAG만으로도 메시지의 구성이 가능하며 메시지 구성의 예제는 그림 3.2와 같다.

TAG/Sentence Message

UdPbC	\g:...,s:...,d:...,n:...,t:...*hh\	\$-----,data,...,data*hh	<CR><LF>
-------	------------------------------------	--------------------------	----------

Sentence Message

UdPbC	\$-----,data,...,data*hh	<CR><LF>
-------	--------------------------	----------

TAG Message

UdPbC	\g:...,s:...,d:...,n:...,t:...*hh\	<CR><LF>
-------	------------------------------------	----------

그림 3.2 TAG/Sentence 메시지 구성의 예

Fig. 3.2 Organization examples of the TAG/Sentence messages

예를 들어 깊이를 나타내는 Sentence Formatter인 "DPT"의 Sentence는 그림 3.3과 같은 형식을 가지는데, 차례대로 '\$'는 Sentence의 시작, "SD"는 음파 장비를 나타내는 장비 식별자, "DPT"는 Sentence Formatter, " $x.x^1$ "은 Water depth(meters)를 나타내는 데이터 필드, " $x.x^2$ "은 Offset(meters)를 나타내는 데이터 필드, " $x.x^3$ "은 Maximum range scale를 나타내는 데이터 필드, "*hh"는 체크섬 필드, "<CR><LF>"는 Sentence의 끝을 나타낸다.

\$SDDPT, $x.x^1$, $x.x^2$, $x.x^3$ *hh	<CR><LF>
--	----------

그림 3.3 DPT Sentence Formatter 구성

Fig. 3.3 Organization Example of the DPT Sentence Formatter

NMEA 0183 버전 4에서는 TAG의 구성을 위해 7개의 인자 코드(parameter code)를 정의하고 있지만 IEC 61162-450에서는 이 중에서 표 3.4에 나타나 있는 5개의 인자 코드만을 사용하며 각 인자 코드에 관한 자세한 설명은 다음과 같다[3, 15].

표 3.4 TAG 블록 인자 코드

Table 3.4 Parameter-codes of the TAG block

코드	파라미터 이름
g	그룹 제어(grouping control)
s	소스 ID(source identification)
d	목적지 ID(destination identification)
n	라인 계수(line-count)
t	텍스트 문자열(text string)

(1) 그룹 제어 : g

'g' 인자 코드는 그룹 제어를 위해 제공되는 코드이며 시스템에서 발생하는 여러 개의 Sentence를 묶어서 그룹으로 송신하도록 해주는 기능을 한다. 'g' 인자 코드를 설정하기 위한 형식은 해당 그룹 번호, 전체 그룹 번호, 그룹 식별 ID의 순서로 구성이 되는데, NMEA 0183 버전 4에는 "1-3-9077"과 같은 형식으로 구성하도록 정의되어 있다. 반면에 IEC 61162-450에는 그룹 식별 ID가 네 자리 숫자에서 두 자리 숫자로 축소된 "1-3-77"과 같은 형식으로 축소하여 구성하도록 정의되어 있다. 또한 'g' 인자 코드가 설정된 TAG/Sentence 메시지는 하나의 TAG 및 하나의 Sentence로 구성하여 송신하도록 정의하며 그림 3.4에서 TAG/Sentence 메시지의 올바른 구성 예제와 올바르지 않은 구성 예를 나타내었다.

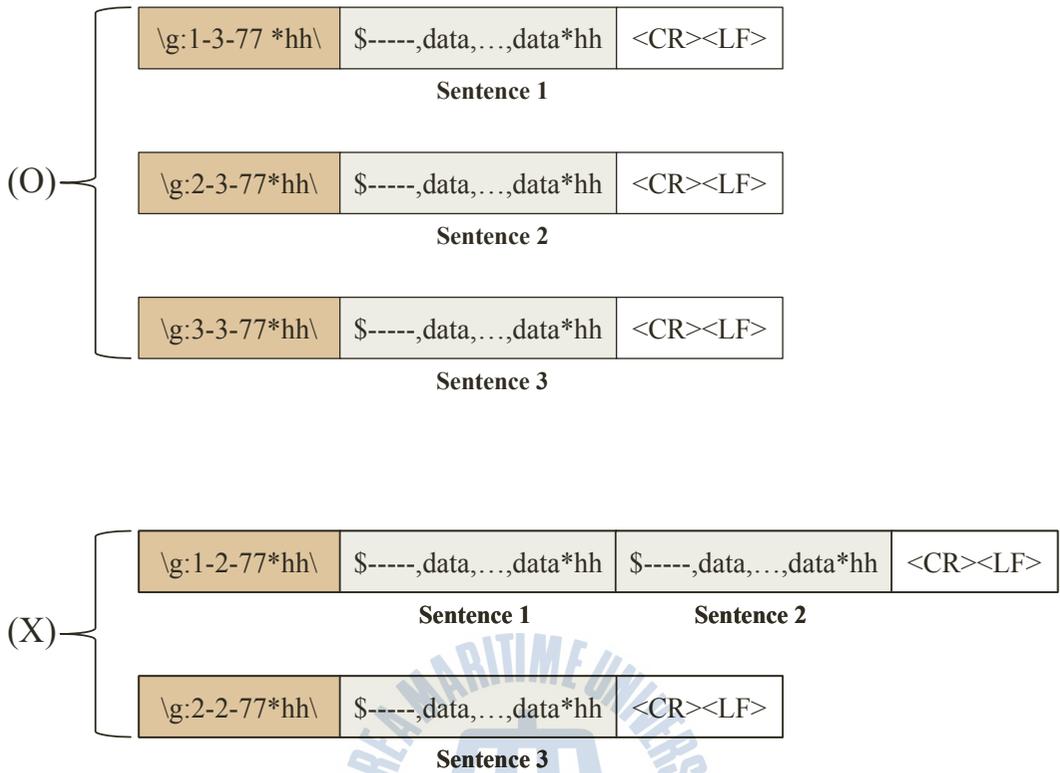


그림 3.4 'g' 인자 코드 구성 예

Fig. 3.4 Organization examples of the 'p' parameter code

(2) 소스 ID : s, 목적지 ID : d

's' 인자 코드는 송신하는 시스템 혹은 장비의 소스 ID를 설정하는 기능을 하며 만약 메시지에 TAG가 포함되면 's' 인자 코드는 필수적인 사항이다. 모든 SF는 SFI(Unique System function ID)라고 하는 두 개의 문자와 네 자리의 숫자 조합의 유일한 ID를 가지며 일반적으로 SFI는 6 바이트의 크기를 가지며 소스 ID에는 최대 15 바이트까지 설정 가능하다. 또한 그림 3.5의 "TI0001", "II0002", "GP0001"과 같이 두 개의 문자는 일반적으로 Talker ID로 부여되고 네 개의 숫자는 0001부터 9999까지 시스템 혹은 장비의 번호에 의해 부여된다. 'd' 인자 코드는 목적지 ID를 설정하는 기능을 하는데 이는 선택 사항이며 's' 인자 코드와 동일한 방식으로 SFI를 이용하여 설정한다.

\s:TI0001*hh\	\$-----,data,...,data*hh	<CR><LF>
---------------	--------------------------	----------

\s:II0002*hh\	\$-----,data,...,data*hh	<CR><LF>
---------------	--------------------------	----------

\d:TI0001,s:GP0001*hh\	\$-----,data,...,data*hh	<CR><LF>
------------------------	--------------------------	----------

그림 3.5 's', 'd' 인자 코드 구성 예

Fig. 3.5 Organization examples of the 's' and 'd' parameter codes

(3) 라인 계수 : n

'n' 인자 코드는 SF에 의해 송신된 TAG/Sentence의 라인 수를 설정하는 기능을 한다. 처음 송신하는 TAG/Sentence에는 1이 부여되고 그 이후의 송신은 1씩 증가된다. 'n' 인자 코드는 1~999 범위의 숫자를 사용하며 그림3.6과 같이 999에 도달하면 다시 1로 재설정된다.

\n:998*hh\	\$-----,data,...,data*hh	<CR><LF>
------------	--------------------------	----------

\n:999*hh\	\$-----,data,...,data*hh	<CR><LF>
------------	--------------------------	----------

\n:1*hh\	\$-----,data,...,data*hh	<CR><LF>
----------	--------------------------	----------

그림 3.6 'n' 인자 코드 구성 예

Fig. 3.6 Organization examples of the 'n' parameter code

(4) 텍스트 문자열 : t

't' 인자 코드는 정해진 형식이 없는 사유(proprietary) 데이터를 송신하는 기능을 하며 구분을 위해 'p' 뒤의 3자리의 제조사 코드가 요구된다. 예를 들어, 제조사 코드가 "hhh"이고 송신하고자하는 사유 데이터가 "check"인 경우에는 `\t:phhhcheck*hh\` 와 같은 형식으로 TAG를 구성한다. 또한 한 TAG에 포함되어 전송 가능한 최대 크기는 61 바이트이다.

3.2.2 바이너리 이미지

바이너리 이미지 메시지의 경우, 메시지는 표 3.5와 같은 형식으로 구성되며 그림 3.7과 같은 여러 정보를 포함하고 있다. 첫 메시지에만 포함되어 송수신되는 디스크립터(descriptor)에는 디스크립터의 길이, 바이너리 이미지의 길이, 인증 상태, 장비 ID, 채널, 데이터 타입의 길이, 데이터 타입, 상태 정보 및 설명 등의 정보가 포함된다. 또한 헤더에는 버전, 보내는 측의 ID, 받는 측의 ID, 타입, 블록 ID, 순차 번호, 최대 순차 번호 등의 정보가 포함된다. 바이너리 이미지 메시지의 크기는 헤더/디스크립터 및 데이터를 포함하여 최대 1472 바이트이며 1472 바이트를 초과하는 바이너리 이미지의 경우 분할하여 송수신한다[3].

표 3.5 바이너리 이미지 메시지 구조

Table 3.5 Structure for the binary image message

항목	설명
"RaUdP" or "RrUdP"	메시지 헤더
null	메시지 헤더 구분자
Header	바이너리 이미지 헤더
Descriptor	바이너리 이미지 디스크립터(첫 메시지에만 포함)
Data	분할된 바이너리 이미지 데이터

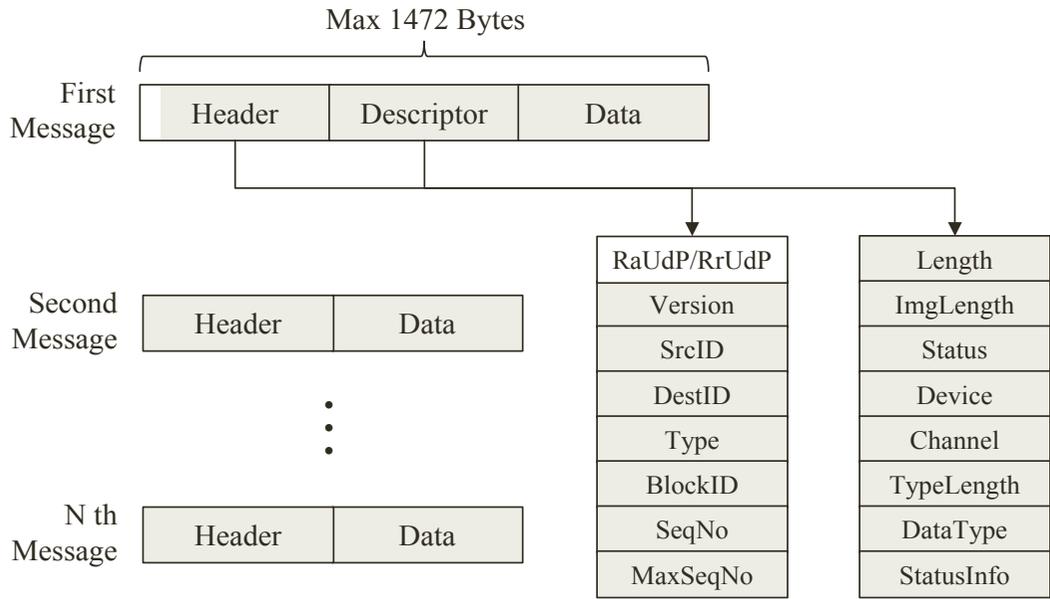


그림 3.7 바이너리 이미지 메시지 내부 구조

Figure 3.7 Internal structure for the binary image message



3.2.3 오류 로그

IEC 61162-450에서는 NF 및 SF에서 발생하는 시스템 로그를 위한 기능을 정의하고 있다. 내부 로그(internal logging)는 NF 및 SF에서 발생한 오류 혹은 특정 사건에 관한 정보를 내부적으로 기록하는 것이다. 또한 외부 로그(external logging)는 오류 혹은 특정 사건에 관한 메시지를 시스템의 외부로 송신하여 로그 서버에 기록하는 것으로 IEC 61162-450에서는 Syslog 프로토콜을 채택하고 있다[16].

내부 로그는 오류 혹은 특정 사건이 발생한 UNIX 시간, 해당하는 메시지, 발생 횟수를 텍스트 파일에 기록하는 방식으로 수행되며 그림 3.8과 같은 구조를 가진다. 또한 외부 로그는 그림 3.9와 같은 구조의 Syslog 메시지를 로그 서버로 송신하고 로그 서버는 수신한 Syslog 메시지를 기록하는 방식으로 수행된다. Syslog 메시지는 오류 코드, 버전, 시간, 호스트 이름, 어플리케이션 이름, 프로세스 ID, 메시지 ID, 구조화된 데이터, 메시지를 포함한다[3].



그림 3.8 내부 로그 메시지 구조

Fig. 3.8 Structure for the internal log message

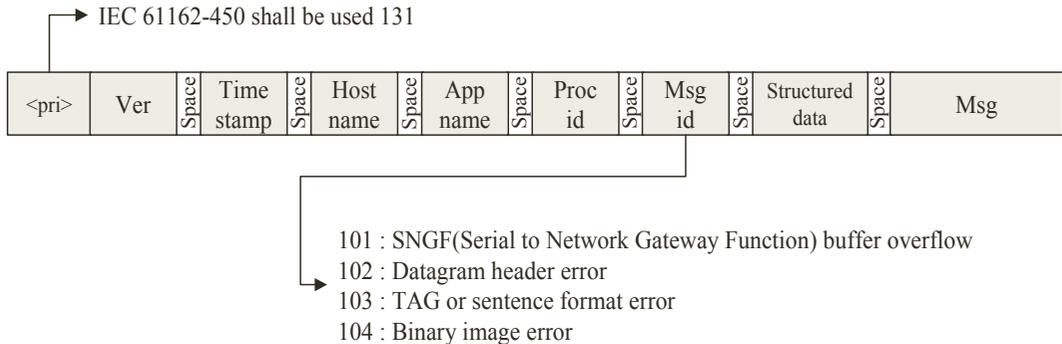


그림 3.9 외부 로그 메시지 구조

Fig. 3.9 Structure for the external log message

제 4 장 선박 통합 네트워크 미들웨어 설계 및 구현

4.1 선박 통합 네트워크 구조 정의

이 논문에서 제안하는 미들웨어의 설계 및 구현을 위한 IEC 61162-450 기반의 선박 통합 네트워크의 구조는 그림 4.1과 같다. 선박 통합 네트워크에는 게이트웨이 어플리케이션, 데이터베이스 어플리케이션, 모니터링 어플리케이션이 존재하며 각 어플리케이션은 NF와 SF를 포함한다.

게이트웨이 어플리케이션은 여러 선박 장비로부터 발생된 정보를 수집하여 IEC 61162-450 표준을 만족하는 형태로 변환한 후 UDP 멀티캐스트 방식으로 송신한다. 또한 모니터링 어플리케이션은 게이트웨이 어플리케이션이 송신한 정보를 수신하여 실시간으로 표시해주는 기능을 하며 데이터베이스 어플리케이션은 게이트웨이 어플리케이션이 송신한 정보를 수신하여 데이터베이스에 저장 및 관리하는 기능을 한다[17].

각 어플리케이션이 포함하고 있는 NF는 선박 통합 네트워크 내에 있는 시스템 간의 상호 연결 및 데이터그램 송수신 등의 기능을 하며 SF는 송수신된 데이터그램의 메시지 처리, 유효성 검사, 오류 로그 등의 기능을 한다.

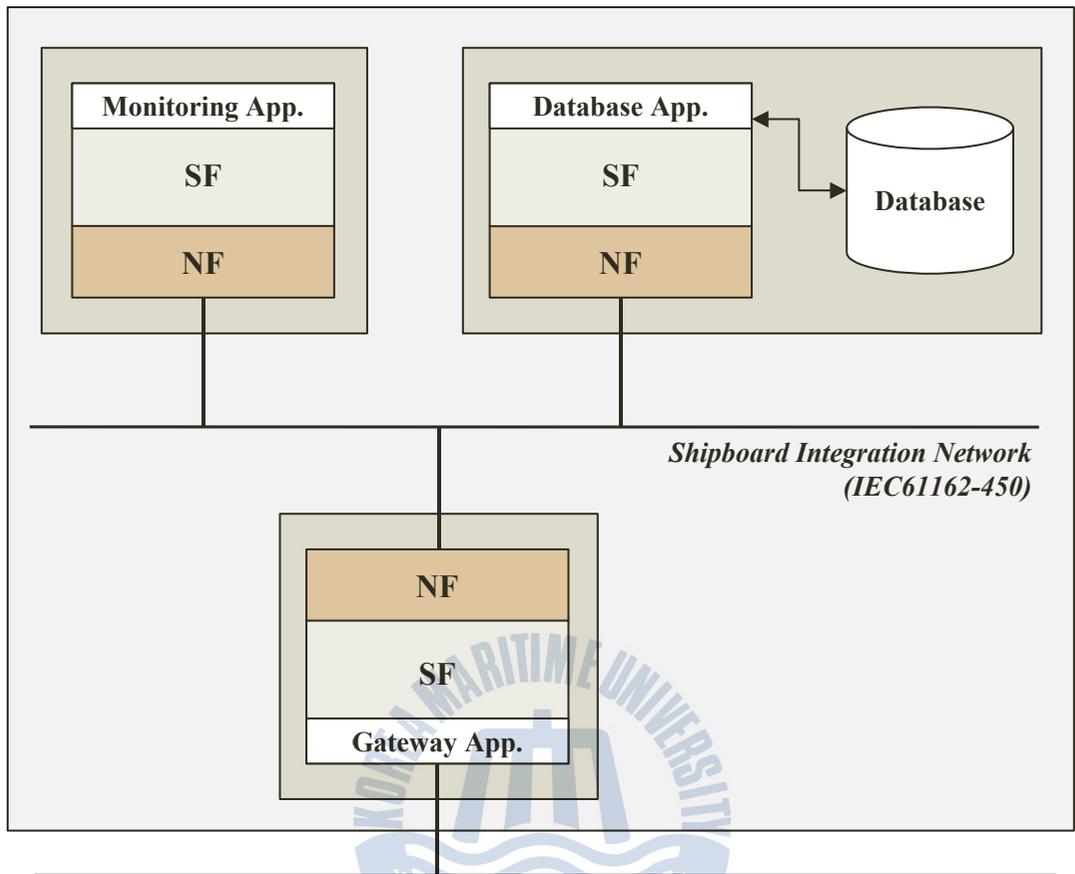


그림 4.1 IEC 61162-450 기반 선박 통합 네트워크

Fig. 4.1 Shipboard integration network based on the IEC 61162-450

4.2 선박 통합 네트워크 미들웨어 설계

IEC 61162-450에는 각 시스템 내의 어플리케이션이 네트워크 기능(NF : Network Function block)과 시스템 기능(SF : System Function block)을 거쳐서 통신을 하도록 정의되어 있다[3]. 따라서 이 절에서는 통신을 위한 NF 모듈 및 SF 모듈을 설계하며 NF 모듈과 SF 모듈의 기능을 이용하기 위한 인터페이스를 제공하는 서비스 모듈을 설계한다.

그림 4.2와 그림 4.3은 선박 통합 네트워크 미들웨어를 구성하는 내부적인 모듈 및 각 송수신 동작 과정을 나타낸다. 미들웨어는 NF 모듈 및 SF 모듈의 Sentence 처리 모듈, 바이너리 이미지 처리 모듈로 구성되어 있으며 각 모듈에서 필요한 버퍼를 포함하고 있다. 또한 선박 통합 네트워크 미들웨어에는 NF 모듈 및 SF 모듈을 이용한 정보 교환 및 정보 처리 서비스를 제공하는 인터페이스인 서비스 모듈(API)이 포함된다.

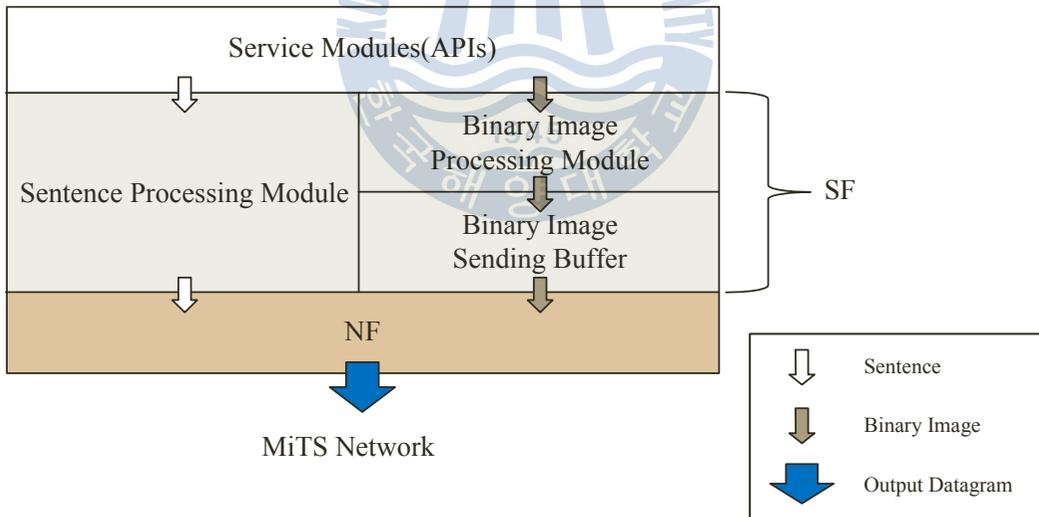


그림 4.2 각 모듈의 동작 흐름도(송신)

Fig. 4.2 Flow diagram of each modules(sending)

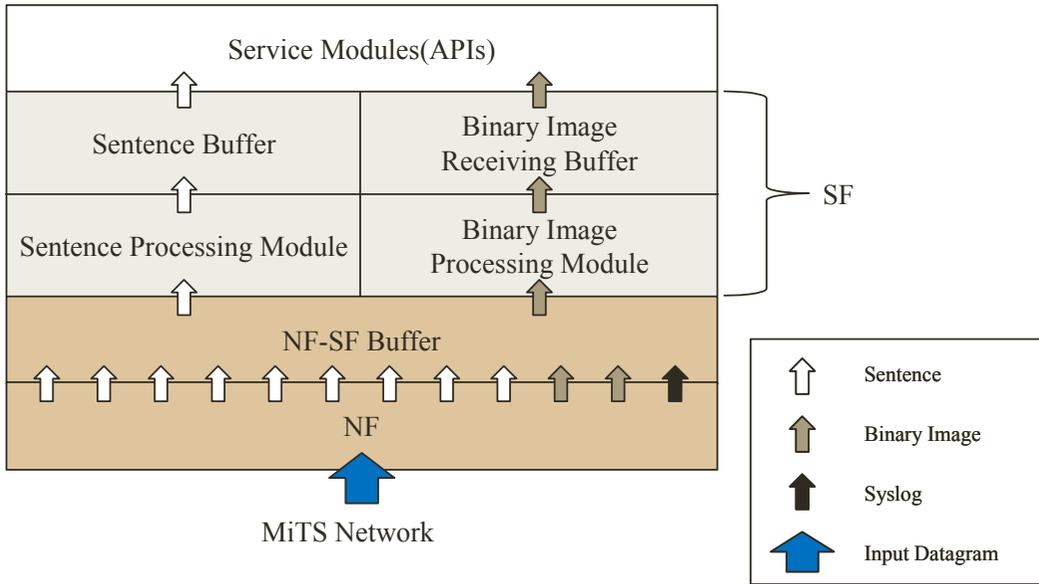


그림 4.3 각 모듈의 동작 흐름도(수신)

Fig. 4.3 Flow diagram of each modules(receiving)

4.2.1 네트워크 기능 모듈

NF 모듈의 주요 기능은 데이터그램의 송수신 기능 및 검사 기능 등이다. 송신을 위해서 NF 모듈은 헤더를 통해 SF로부터 전달받은 메시지가 TAG/Sentence 인지 바이너리 이미지인지를 구별한다. 그 뒤, 데이터그램을 만들어 해당하는 IP 주소 및 포트에 UDP 멀티캐스트 방식을 이용하여 송신한다.

수신을 위해서 NF 모듈은 데이터그램의 수신 대기(listening) 쓰레드(thread)를 해당 IP 주소 및 포트 번호 별로 생성하고 모든 쓰레드를 활성화하여 수신한다. 이때, TAG/Sentence를 위한 쓰레드 9개와 바이너리 이미지를 위한 쓰레드 2개, Syslog를 위한 쓰레드 1개가 생성 및 활성화된다.

데이터그램이 수신되면 데이터그램 검사를 수행한다. 먼저 수신된 데이터그램의 크기가 1472 바이트를 초과하지 않는지를 검사하는데 만약 1472 바이트를 초과하면, 그 데이터그램을 무시한다. 그 뒤 UDP 체크섬 필드의 설정 여부를 확인하는데 만약 체크섬 필드가 0이면, 체크섬 필드가 설정되지 않은 것이므로 올바른 데이터그램이 아닌 것으로 판단하고 그 데이터그램을 무시한다.

여러 데이터그램이 동시에 수신되는 경우도 있기 때문에 소실 방지 및 처리를 위해 유효성 검사에 위배되지 않는 데이터그램을 NF-SF 버퍼에 저장한다. 이 버퍼는 큐의 형태로 구현되어 있으며 처리는 선입선출(First In First Out) 방식으로 처리된다. 또한 SF에서 이 버퍼를 계속 감시하여 데이터그램이 저장되면 SF로 가져오고 저장되어 있던 데이터그램을 버퍼에서 삭제하는 방식으로 버퍼를 관리한다.

4.2.2 시스템 기능 모듈

(1) Sentence 처리 모듈

Sentence 처리 모듈의 주요 기능은 IEC 61162-450 메시지 헤더 추가/제거 기능, TAG/Sentence 유효성 검사 및 처리 기능 등이다. 송신을 위해서, Sentence 처리 모듈은 어플리케이션으로부터 전달받은 메시지의 처리를 한 후 헤더를 추가하여 NF 모듈로 전달한다.

수신을 위해서, Sentence 처리 모듈은 NF-SF 버퍼에 저장되어 있는 데이터그램의 헤더를 확인하여 TAG/Sentence 메시지면 Sentence 처리 모듈로 가져오고 헤더를 제거한다. 그 뒤, 그림 4.4와 같은 과정으로 유효성 검사를 하며 위배되지 않으면 수신된 Sentence를 Sentence Formatter에 따라 미리 할당되어 있는 Sentence 버퍼에 덮어쓴다. 이때, 체크섬 검사는 '*' 이후의 두 자리 16진수를 이용하며 '\$'나 '!'의 이후부터 '*' 전까지의 문자들을 시작/끝 비트가 없는 8비트 XOR(exclusive OR) 연산의 결과 값을 통해 수행된다.

1. 최대 길이를 넘지 않는가?
 - 1-1. TAG가 80 바이트 ('\ ' 포함)를 초과하지 않는가?
 - 1-2. Sentence가 82 바이트 ('\$', '<CR>', '<LF>' 포함)를 넘지 않는가?
2. TAG/Sentence가 '\$' 혹은 '!' 혹은 '\ '로 시작하는가?
3. TAG에 올바른 인자 코드가 있는가?
4. Sentence에 올바른 Talker ID 및 Sentence Formatter가 있는가?
5. 사용될 수 없는 문자가 사용되지 않았는가?
 - 5-1. 16진수 20~7F까지 범위 내의 유효 문자인가?
 - 5-2. ', '를 제외한 예약 문자가 오지 않았는가?
6. '*'의 이후에 두 자리 16 진수 체크섬이 있는가?
7. 체크섬 오류가 없는가? (8비트 XOR 연산)
8. TAG/Sentence가 '<CR>', '<LF>'로 끝나는가?

그림 4.4 TAG/Sentence 메시지의 유효성 검사 알고리즘

Fig. 4.4 Validity check algorithms for the TAG/Sentence messages

Sentence 버퍼는 그림 4.5와 같은 구조로 되어있다. "DPT", "MWD", "MTW", "RMC" 등 각 Sentence Formatter에 따라 하나씩의 저장 공간이 할당되어 있으며 다중 Sentence를 지원하는 "TXT" 및 여러 형식의 정보에 사용될 수 있는 "XDR"은 그 특성에 따른 처리를 위하여 여러 개의 저장 공간이 할당되어 있다. 또한 Sentence Formatter에 상관없이 수신된 TAG/Sentence, Sentence, TAG를 순차적으로 저장하기 위한 공간도 할당되어 있다. 여러 어플리케이션은 서비스 모듈이 제공하는 인터페이스를 통해 이 버퍼를 참조하여 원하는 데이터를 얻을 수 있다.

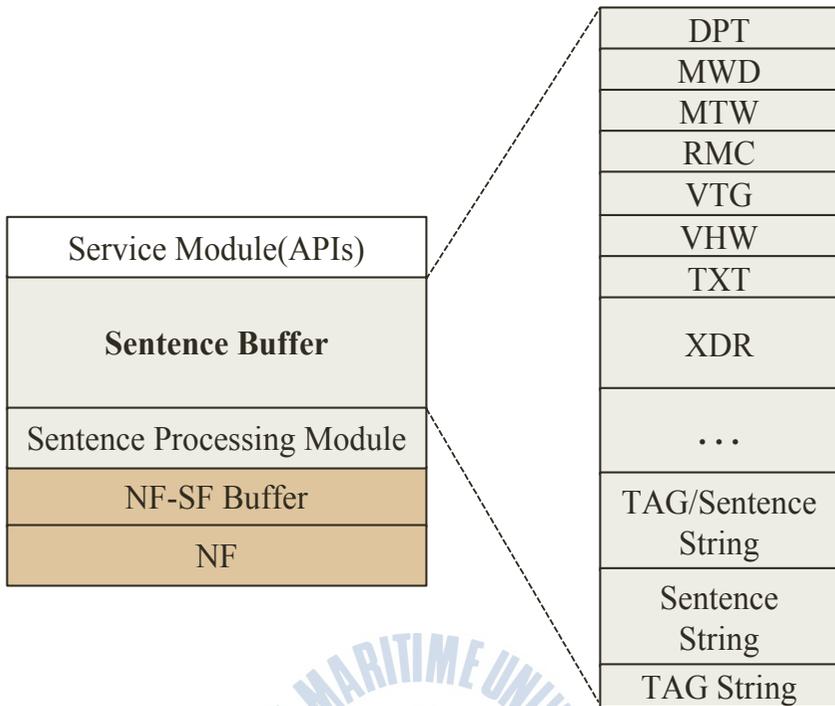


그림 4.5 Sentence 버퍼 구조
 Fig. 4.5 Structure of the Sentence buffer

(2) 바이너리 이미지 처리 모듈

바이너리 이미지 처리 모듈은 TAG/Sentence 메시지가 아닌 바이너리 이미지를 송수신하는데 사용되는 모듈이다. 이 모듈의 주요 기능은 헤더 생성 기능, 디스크립터 생성 기능, 바이너리 이미지 데이터 분할 및 조합 기능 등이 있다.

송신의 경우, 그림 4.6과 같은 순서로 동작한다. 먼저 바이너리 이미지 모듈을 생성하고 초기화 한 후 헤더 및 디스크립터를 만든다. 만약 헤더와 디스크립터를 포함한 바이너리 이미지가 1472 바이트를 넘는다면, 바이너리 이미지를 1472 바이트 단위로 분할한다. 분할된 바이너리 이미지는 송신 버퍼에 저장되어 순차적으로 하나씩 송신된다. 이때, 헤더와 디스크립터에 포함되는 여러 정보 중에 블록 ID는 수신 측에서 동일한 이미지인지 아닌지를 식별하기 위한 것으로 최초에는 정수를 무작위(random)로 생성하여 부여하고 그 이후의 이미지

에 대해서는 생성되었던 정수에 1을 더하는 방식으로 부여한다. 또한 순차 번호는 분할된 파일이 몇 번째의 파일인지를 식별하기 위해 0부터 1씩 증가하는 방식으로 부여되며 순차 번호가 최대 순차 번호와 같아지면 모두 전송되었다고 판단한다.

1. 송신할 바이너리 이미지의 헤더와 디스크립터를 생성한다.
2. 1472 바이트를 초과하면, 바이너리 이미지를 1472 바이트 단위로 분할하여 순서대로 송신 버퍼에 저장한다.
 - 2-1. 첫 번째 데이터그램에는 헤더와 디스크립터, 바이너리 이미지를 포함하여 분할 저장한다.
 - 2-2. 두 번째 데이터그램 부터는 디스크립터를 제외하고 분할 저장한다.
3. 송신 버퍼에서 데이터그램을 가져온다.
4. 순차 번호를 설정한다. (0부터 1씩 증가)
5. 헤더의 나머지 필드를 설정한다.
6. 데이터그램을 송신한다.
 - 6-1. 모든 데이터그램이 송신되지 않았으면, 3단계부터 실행한다.
 - 6-2. 모두 송신 되었으면, 1단계부터 실행한다.

그림 4.6 바이너리 이미지의 송신 알고리즘

Fig. 4.6 Sending algorithms for the binary images

또한 수신인 경우, 그림 4.7과 같은 순서로 동작하는데 NF-SF 버퍼에 저장된 메시지 헤더를 보고 구별하여 바이너리 이미지 처리 모듈로 가져온다. 바이너리 이미지의 헤더와 디스크립터를 확인하여 만약 분할된 바이너리 이미지라면, 수신 버퍼에 저장하고 순차 번호를 확인하여 나머지가 모두 수신되면 조합한다.

1. 데이터그램을 수신하여 헤더와 디스크립터를 분리한다.
2. 수신한 데이터그램 헤더의 블록 ID와 이전에 수신하여 수신 버퍼에 저장된 데이터그램의 블록 ID를 비교한다.
 - 2-1. 블록 ID가 같으면, 이어서 수신 버퍼에 분할된 바이너리 이미지를 저장한다.
 - 2-2. 블록 ID가 다르면, 수신 버퍼를 초기화한 후 1단계부터 실행한다.
3. 저장된 데이터그램의 순차 번호와 최대 순서 번호를 비교한다.
 - 3-1. 동일하면, 데이터그램 수신이 완료된 것이므로 버퍼에 저장된 바이너리 이미지를 완성시켜서 어플리케이션으로 전달한 후 수신 버퍼를 초기화한다.
 - 3-2. 동일하지 않으면, 1단계부터 실행한다.

그림 4.7 바이너리 이미지의 수신 알고리즘

Fig. 4.7 Receiving algorithms for the binary images

4.2.3 서비스 모듈

선박 통합 네트워크에서 시스템 간의 상호 정보 교환을 위해 제공되어야 할 서비스 요구사항의 도출을 위해서는 첫째, 송신을 위한 시스템인지, 수신을 위한 시스템인지를 고려해야 한다. 그림 4.1의 게이트웨이 어플리케이션은 송신 기능이 필요하고 모니터링 어플리케이션이나 데이터베이스 어플리케이션 등은 수신 기능이 필요하다. 둘째, 송수신 서비스를 위한 메시지의 형식이 TAG/Sentence인지, 바이너리 이미지인지를 구분해야 하며 만약 TAG/Sentence 메시지인 경우에는 추가적으로 TAG/Sentence, Sentence, TAG를 구분해야 한다. 셋째, TAG/Sentence 메시지의 형식이 일반적인 문자열(string) 형식인지, 선박 통합 네트워크 내에 있는 시스템 구성의 편의성을 위해 정의된 구조체(struct) 형식인지를 구분해야 한다. 도출된 서비스 요구사항에 따라 정의된 서비스 모듈의 API 함수는 표 4.1과 같다[18].

표 4.1 서비스 모듈 API 함수

Table 4.1 API functions of the service module

	함수 이름	설 명
1	SendTAGSentence	TAG/Sentence 송신 (구조체 형식)
2	SendTAGSentenceString	TAG/Sentence 송신 (문자열 형식)
3	SendSentence	Sentence 송신 (구조체 형식)
4	SendSentenceString	Sentence 송신 (문자열 형식)
5	SendTAG	TAG 송신 (구조체 형식)
6	SendTAGString	TAG 송신 (문자열 형식)
7	GetTAGSentence	TAG/Sentence 수신 (구조체 형식)
8	GetTAGSentenceString	TAG/Sentence 수신 (문자열 형식)
9	GetSentence	Sentence 수신 (구조체 형식)
10	GetSentenceString	Sentence 수신 (문자열 형식)
11	GetTAG	TAG 수신 (구조체 형식)
12	GetTAGString	TAG 수신 (문자열 형식)
13	SendBinary	Binary Image 송신
14	GetBinary	Binary Image 수신

4.3 선박 통합 네트워크 미들웨어 구현

4.3.1 데이터 형식 정의

선박 통합 네트워크를 위한 미들웨어의 구현을 위하여 NMEA 0183 버전 4에 정의되어 있는 100여개의 Sentence Formatter 중 8개의 Sentence Formatter를 선정하였다. 각 Sentence Formatter의 의미는 표 4.2와 같으며 그림 4.8과 같이 구조체 형식으로 정의하였다.

표 4.2 Sentence Formatter 및 의미

Table 4.2 Sentence Formatters and its meanings

Sentence Formatter	설명
DPT	깊이
MTW	수온
MWD	풍향, 풍속
RMC	권고된 최소 명세의 GNSS 데이터(GPS 데이터)
TXT	텍스트 전송
VHW	물의 속도, 헤딩
VTG	방향(Course Over Ground), 실속력(Speed Over Ground)
XDR	트랜스듀서 측정치

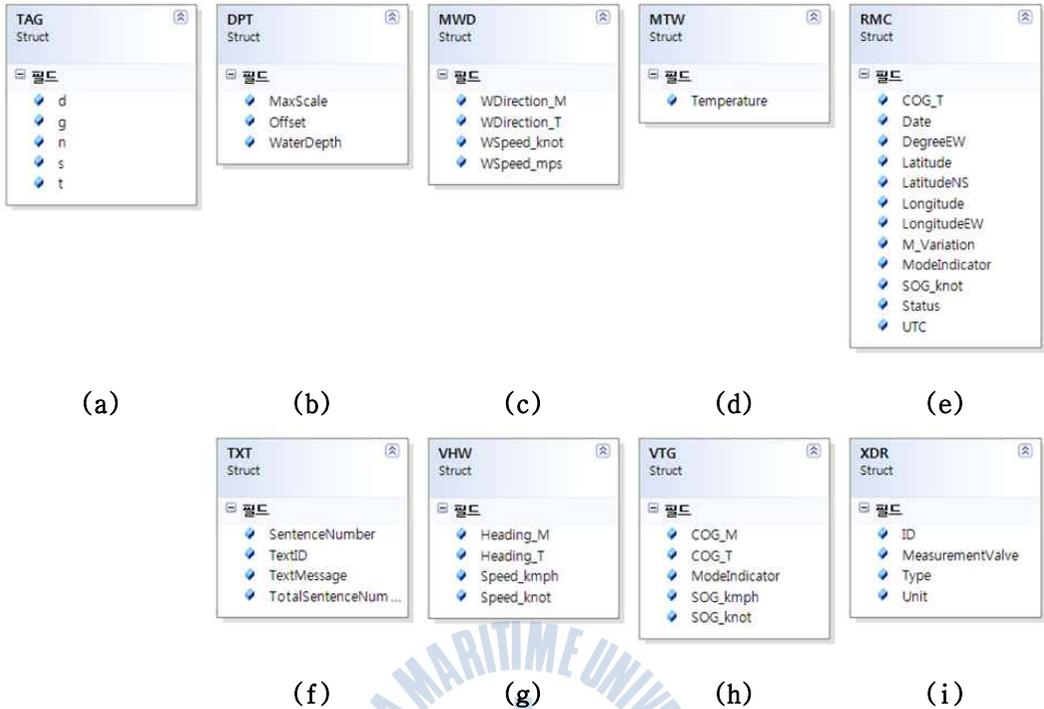


그림 4.8 TAG 및 Sentence Formatter 구조체
 Fig. 4.8 Data structures of the TAG and Sentence Formatter

4.3.2 클래스 구현

설계한 내용을 바탕으로 구현된 선박-통합 네트워크 미들웨어는 그림 4.9와 같이 4개의 클래스로 구성된다. 먼저 LSI 클래스는 인터페이스를 제공하는 클래스로 어플리케이션에게 NetworkFunction 클래스, SentenceProcessingModule 클래스, BinaryProcessingModule 클래스가 포함하는 기능을 제공하기 위하여 각 클래스의 포인터를 가진다. 또한 NetworkFunction 클래스는 UDP 소켓 통신을 통한 데이터그램의 송수신 및 데이터그램 검사 기능 등을 위한 여러 멤버 변수와 멤버 함수를 포함한다. SentenceProcessingModule 클래스는 TAG/Sentence 메시지 처리 및 TAG/Sentence 유효성 검사 기능 등을 위한 여러 멤버 변수와 멤버 함수를 포함한다. BinaryProcessingModule 클래스는 바이너리 이미지의 처리를 위한 멤버 변수 및 멤버 함수를 포함한다.

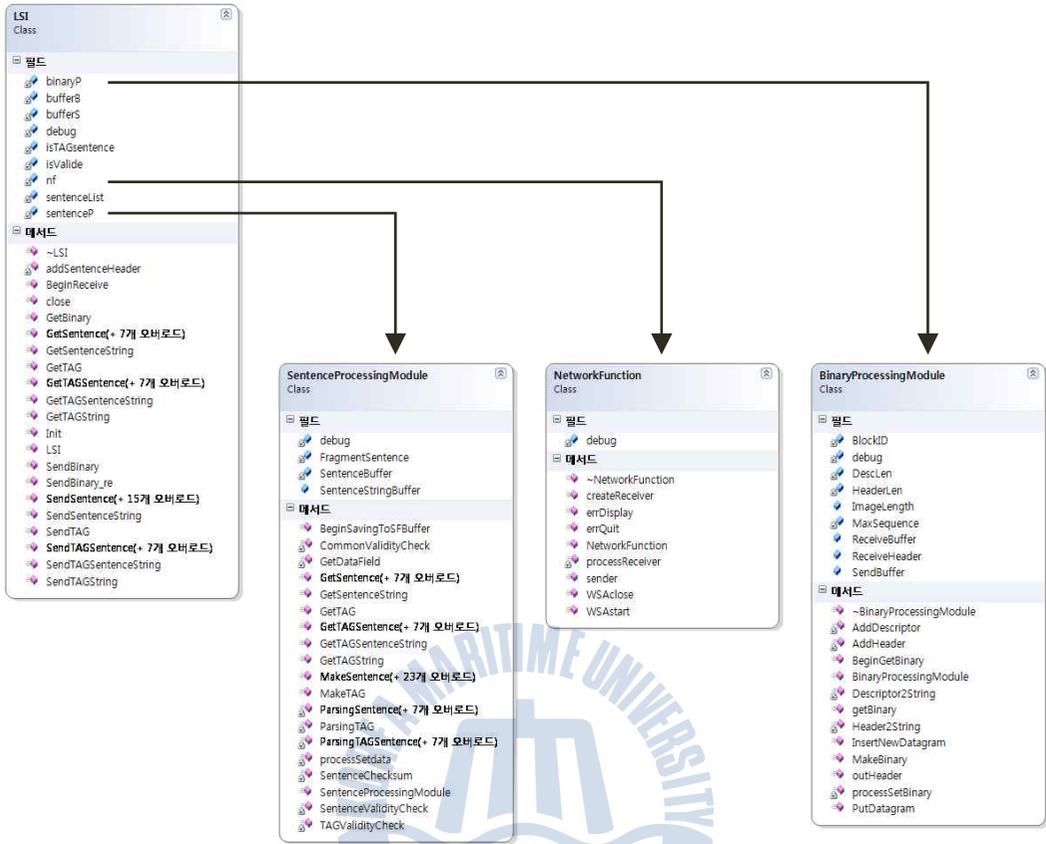


그림 4.9 선박 통합 네트워크 미들웨어의 클래스 다이어그램

Fig. 4.9 Class diagrams of the shipboard integration network middleware

그림 4.10은 API 함수 호출 시 내부 함수의 동작 과정의 예를 나타내는데, 시스템의 어플리케이션이 SendTAGSentence 함수를 호출하면 내부적으로 IEC 61162-450 표준을 만족하는 TAG/Sentence로 변환해주는 함수인 MakeTAGSentence 함수가 호출되어 송신한다. 다른 시스템에서 GetTAGSentence 함수를 호출하면 NF-SF 버퍼에 저장된 TAG/Sentence를 파싱(parsing)하여 구조체로 변환해주는 함수인 ParsingTAGSentence 함수가 내부적으로 호출되어 TAG/Sentence가 구조체 형식으로 어플리케이션에 전달된다. GetTAGSentenceString 함수는 버퍼에서 파싱 없이 바로 문자열 형식으로 어플리케이션에 전달된다.

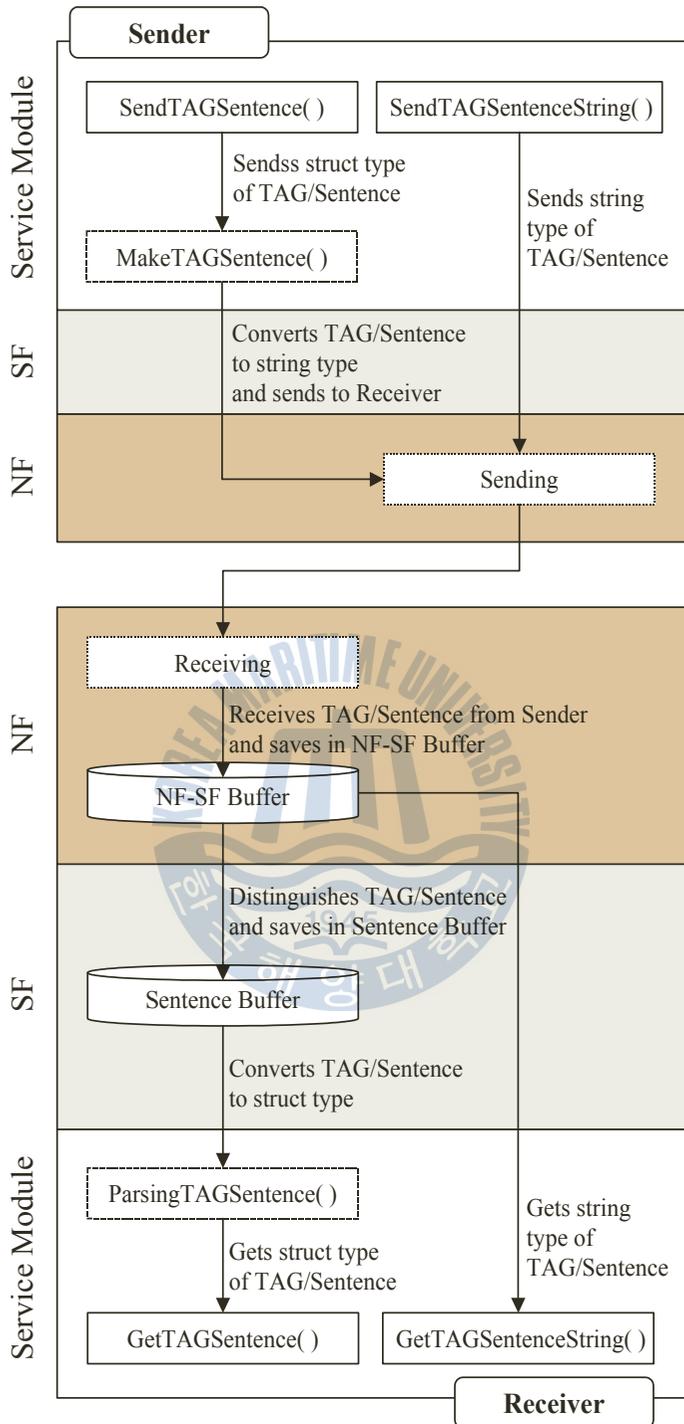


그림 4.10 API 함수 호출시 내부 동작의 예

Fig. 4.10 Example of the internal operation when API function is called

제 5 장 선박 통합 네트워크 미들웨어 시험

5.1 시험 요구사항

IEC 61162-450에서는 기본적으로 성능 검사(performance check)를 위해서는 데이터그램 및 메시지의 성공적인 송수신 여부를 확인하도록 하고 있으며 성능 실험(performance test)을 위해서는 여러 실험 시나리오 상에서의 성능 평가를 하도록 요구하고 있다.

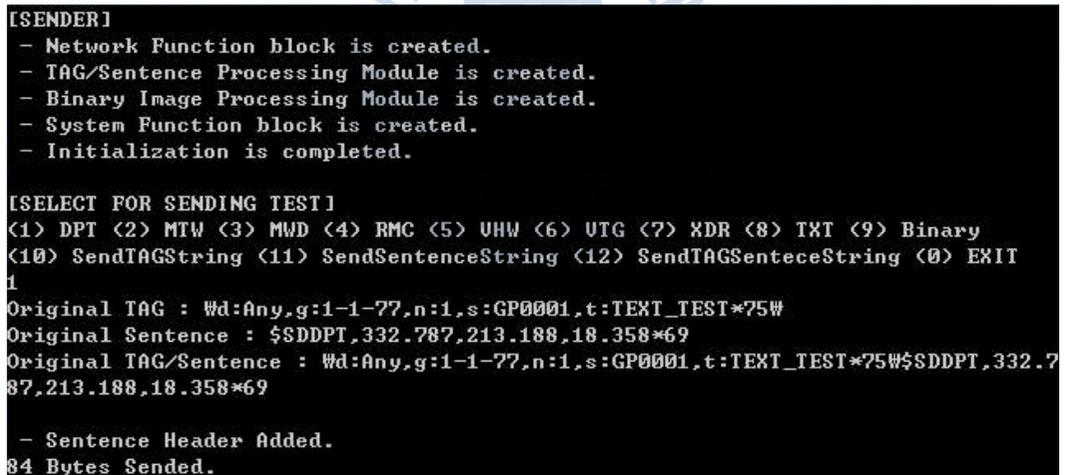
또한 IEC 61162-450에서는 이 표준에 정의된 형식을 올바르게 만족하는 송수신인지 혹은 그렇지 않은 송수신인지를 실험하기 위한 여러 항목을 정의하고 있으며 정의된 각 항목에 대해서는 NF 및 SF를 구성하는 각 모듈의 유효성 검사 과정에서 표준을 만족하는지 혹은 표준을 만족하지 않는지를 검사 및 판단하게 된다.

실험을 위하여 개발한 미들웨어를 라이브러리 파일로 생성(export)하여 Sender 어플리케이션 및 Receiver 어플리케이션에 포함(import)하는 방법으로 Sender 어플리케이션 및 Receiver 어플리케이션을 구현하였다. 또한 네트워크에 연결된 두 대의 컴퓨터에서 TAG/Sentence 및 바이너리 이미지가 UDP 멀티캐스트 방식으로 송수신 가능한지와 Syslog 기능이 올바르게 동작하는지를 실험하였다.

5.2 선박 통합 네트워크 미들웨어 시험

5.2.1 TAG/Sentence 송수신

TAG/Sentence 송수신의 실험을 위해서 올바른 TAG/Sentence 메시지인 경우와 그렇지 않은 경우의 시나리오로 시험해 보았다. Sender 어플리케이션에서 올바른 TAG/Sentence 메시지를 송신하고 Receiver 어플리케이션에서는 구조체 형식으로 TAG/Sentence 메시지를 수신하는 GetTAGSentence 함수와 문자열 형식으로 TAG/Sentence를 수신하는 GetTAGSentenceString 함수의 호출을 통해 메시지를 수신하였다. 송수신된 메시지의 크기가 84 바이트로 동일하다는 것과 내용이 "Wd:Any,g:1-1-77,n:1,s:GP0001,t:TEXT_TEST*75W\$SDDPT,332.787,213.188,18,358*69"로 동일하다는 것을 확인할 수 있으며 유효성 검사를 오류 없이 통과하는 것을 그림 5.1과 그림 5.2에서 확인할 수 있다.



```
[SENDER]
- Network Function block is created.
- TAG/Sentence Processing Module is created.
- Binary Image Processing Module is created.
- System Function block is created.
- Initialization is completed.

[SELECT FOR SENDING TEST]
<1> DPT <2> MTW <3> MWD <4> RMC <5> UHW <6> UTG <7> XDR <8> TXT <9> Binary
<10> SendTAGString <11> SendSentenceString <12> SendTAGSenteceString <0> EXIT
1
Original TAG : Wd:Any,g:1-1-77,n:1,s:GP0001,t:TEXT_TEST*75W
Original Sentence : $SDDPT,332.787,213.188,18.358*69
Original TAG/Sentence : Wd:Any,g:1-1-77,n:1,s:GP0001,t:TEXT_TEST*75W$SDDPT,332.7
87,213.188,18.358*69

- Sentence Header Added.
84 Bytes Sended.
```

그림 5.1 TAG/Sentence 메시지 송신 시험

Fig. 5.1 Sending test for the TAG/Sentence message

```

[RECEIUER]
- Network Function block is created.
- TAG/Sentence Processing Module is created.
- Binary Image Processing Module is created.
- System Function block is created.
- Initialization is completed.
- Start listening Datagram.
- TAG/Sentence Processing Thread Creation.

84 Bytes Received.
Received Datagram :
UdPbC Wd:Any,g:1-1-77,n:1,s:GP0001,t:TEXT_TEST*75W$SDDPT,332.787,213.188,18.358*
69

- Sentence Validity Check Passed(SUC).

- Sentence Validity Check Passed.

GetTAGSentence :
TAG> 1-1-77 Any GP0001 1 TEXT_TEST
Sentence> 213.188 332.787 18.358
GetTAG/SentenceString : Wd:Any,g:1-1-77,n:1,s:GP0001,t:TEXT_TEST*75W$SDDPT,332.7
87,213.188,18.358*69

```

그림 5.2 TAG/Sentence 메시지 수신 시험

Fig. 5.2 Receiving test for the TAG/Sentence message

만약 그림 5.3과 같이 Sender 어플리케이션에서 고의적으로 오류를 발생시킨 Sentence 메시지를 송신하면, Receiver 어플리케이션에서는 그림 5.4와 같이 Sentence 크기 오류 및 예약 문자 오류가 발생하는 것을 확인할 수 있다. 이 때, 최대 크기가 82 바이트를 초과하지 못하도록 정의되어 있는 Sentence 메시지가 128 바이트의 크기로 송수신되었기 때문에 Sentence 크기 오류인 "Sentence Size Error"가 발생하였다. 또한 캡슐화된 Sentence의 시작에만 올 수 있는 문자인 '!'가 수신된 Sentence의 데이터 필드에 포함되어 송수신되었기 때문에 예약 문자 오류인 "Reserved Character Error"가 발생하였다.

5.2.2 바이너리 이미지 송수신

그림 5.5와 같이 Sender 어플리케이션에서 전체 크기가 13387 바이트인 바이너리 이미지를 송신하면, 바이너리 이미지의 디스크립터 및 헤더를 생성 및 추가하고 1472 바이트 단위로 분할한 후 송신한다. 이때, 바이너리 이미지는 10개로 분할되어 송신 버퍼에 저장되며 0에서부터 9까지의 순차 번호에 따라 차례대로 송신하는 것을 볼 수 있다.

```
[SELECT FOR SENDING TEST]
<1> DPT <2> MTW <3> MWD <4> RMC <5> UHW <6> UTG <7> XDR <8> TXT <9> Binary
<10> SendTAGString <11> SendSentenceString <12> SendTAGSenteceString <0> EXIT
9
ImageLength : 13387
- Descriptor Added.
MaxSequenceSize:9
- Header Added.
1472 Bytes Sended.
503 Bytes Sended.
```

그림 5.5 바이너리 이미지 송신 시험

Fig. 5.5 Sending test for the binary image

그림 5.6과 같이 Receiver 어플리케이션에서는 수신한 바이너리 이미지의 헤더 및 디스크립터 정보를 출력하고 바이너리 이미지를 수신 버퍼에 저장한다. 그 뒤, 분할된 바이너리 이미지가 모두 수신되면 처리한 후 "소스 ID_블록 ID"의 형태로 구성된 파일 이름으로 저장하는 것을 볼 수 있다.

```
[Header Information]
1472 Bytes Received.
1472 Bytes Received.
Token :
RaUDP
Version : 1
Source ID : cc0001
Destination ID : XXXXXX
Type : 1
Block ID : 25873
Sequence Num : 0
Max Sequence : 9

[Descriptor Information]
Length : 24
ImageLength : 13387
StatusOfAcquistition : 0
Device : 1
Channel : 1
TypeLength : 10

1472 Bytes Received.
503 Bytes Received.
Get Binary (file name) :
cc0001_25873.jpg
```

그림 5.6 바이너리 이미지 수신 시험
Fig. 5.6 Receiving test for the binary image

5.2.3. 오류 로그

그림 5.7은 내부 로그를 텍스트 파일에 저장한 모습이며 앞서 3장에서 제시한 그림 3.8과 같이 내부 로그에는 오류 혹은 특정 사건이 발생한 시간(unix time), 해당하는 메시지, 발생 수가 기록된다. 올바르지 않은 Sentence 송수신을 나타낸 그림 5.3과 그림 5.4의 예약 문자 오류와 Sentence 크기 오류로 인하여 Sentence 유효성 검사가 실패했다는 내용이 기록된다.

이때, 첫 번째 줄의 "Reserved Character Error"는 예약 문자 오류가 1회 발생했다는 의미이며 두 번째 줄의 "Sentence Validity Check Failed"는 Sentence 유효성 검사 실패가 1회 발생했다는 의미이다. 또한 세 번째 줄의 "Sentence Size Error"는 Sentence 크기 오류가 1회 발생했다는 의미이며 네 번째 줄의 "Sentence Validity Check Failed"는 유효성 검사 실패가 2회 발생했다는 의미이다. 그 이하의 줄도 마찬가지로 오류 혹은 특정 사건의 메시지 및 발생 수가 기록된다.

1292827543	Message : Reserved Character Error(SVC)	Total Occurrence Count :	1
1292827543	Message : Sentence Validity Check Failed	Total Occurrence Count :	1
1292827544	Message : Sentence Size Error(SVC)	Total Occurrence Count :	1
1292827544	Message : Sentence Validity Check Failed	Total Occurrence Count :	2
1292827544	Message : Reserved Character Error(SVC)	Total Occurrence Count :	2
1292827544	Message : Sentence Validity Check Failed	Total Occurrence Count :	3
1292827545	Message : Sentence Size Error(SVC)	Total Occurrence Count :	2
1292827545	Message : Sentence Validity Check Failed	Total Occurrence Count :	4
1292827545	Message : Reserved Character Error(SVC)	Total Occurrence Count :	3
1292827545	Message : Sentence Validity Check Failed	Total Occurrence Count :	5
1292827545	Message : Sentence Size Error(SVC)	Total Occurrence Count :	3
1292827545	Message : Sentence Validity Check Failed	Total Occurrence Count :	6
1292827546	Message : Reserved Character Error(SVC)	Total Occurrence Count :	4
1292827546	Message : Sentence Validity Check Failed	Total Occurrence Count :	7
1292827546	Message : Sentence Size Error(SVC)	Total Occurrence Count :	4
1292827546	Message : Sentence Validity Check Failed	Total Occurrence Count :	8
1292827547	Message : Reserved Character Error(SVC)	Total Occurrence Count :	5
1292827547	Message : Sentence Validity Check Failed	Total Occurrence Count :	9
1292827547	Message : Sentence Size Error(SVC)	Total Occurrence Count :	5
1292827547	Message : Sentence Validity Check Failed	Total Occurrence Count :	10
1292827548	Message : Reserved Character Error(SVC)	Total Occurrence Count :	6
1292827548	Message : Sentence Validity Check Failed	Total Occurrence Count :	11
1292827548	Message : Sentence Size Error(SVC)	Total Occurrence Count :	6
1292827548	Message : Sentence Validity Check Failed	Total Occurrence Count :	12
1292827548	Message : Reserved Character Error(SVC)	Total Occurrence Count :	7
1292827548	Message : Sentence Validity Check Failed	Total Occurrence Count :	13
1292827556	Message : Sentence Size Error(SVC)	Total Occurrence Count :	7
1292827556	Message : Sentence Validity Check Failed	Total Occurrence Count :	14

그림 5.7 내부 로그 저장 결과

Fig. 5.7 Saving results of the internal logs

그림 5.8은 외부 로그를 텍스트 파일에 저장한 모습이며 외부 로그는 시스템에서 발생한 오류 혹은 특정 사건의 정보를 로그 서버로 송신하는 방식으로 수행된다. 앞서 3장에서 제시한 그림 3.9와 같이 외부 로그에는 오류 코드, 버전, 시간, 호스트 이름, 어플리케이션 이름, 프로세스 ID, 메시지 ID, 구조화된 데이터, 메시지가 기록된다.

첫 번째 줄의 예를 들면, 차례대로 "<131>"은 IEC 61162-450에서 사용되는 pri 코드, "1"은 버전, "1292827543"은 UNIX 시간, "H2G"는 호스트 이름, "receive_test"는 어플리케이션 이름, "11"은 프로세스 ID, "103"은 메시지 ID, "-"는 구조화된 데이터가 없다는 기호, "Reserved Character Error(SVC)"는 메시지이다. 그 이하의 줄도 마찬가지로의 형식으로 외부 로그가 기록된다.

<131>	1	1292827543	H2G	receive_test	11	103	-	Reserved Character Error(SVC)
<131>	1	1292827544	H2G	receive_test	11	103	-	Sentence Size Error(SVC)
<131>	1	1292827544	H2G	receive_test	11	103	-	Reserved Character Error(SVC)
<131>	1	1292827545	H2G	receive_test	11	103	-	Sentence Size Error(SVC)
<131>	1	1292827545	H2G	receive_test	11	103	-	Reserved Character Error(SVC)
<131>	1	1292827545	H2G	receive_test	11	103	-	Sentence Size Error(SVC)
<131>	1	1292827546	H2G	receive_test	11	103	-	Reserved Character Error(SVC)
<131>	1	1292827546	H2G	receive_test	11	103	-	Sentence Size Error(SVC)
<131>	1	1292827547	H2G	receive_test	11	103	-	Reserved Character Error(SVC)
<131>	1	1292827547	H2G	receive_test	11	103	-	Sentence Size Error(SVC)
<131>	1	1292827548	H2G	receive_test	11	103	-	Reserved Character Error(SVC)
<131>	1	1292827548	H2G	receive_test	11	103	-	Sentence Size Error(SVC)
<131>	1	1292827548	H2G	receive_test	11	103	-	Reserved Character Error(SVC)
<131>	1	1292827556	H2G	receive_test	11	103	-	Sentence Size Error(SVC)

그림 5.8 외부 로그 저장 결과

Fig. 5.8 Saving results of the external logs

5.3 선박 통합 네트워크 시험

개발한 선박 통합 네트워크 미들웨어의 검증을 위하여 4장의 그림 4.1에서 제시했던 선박 통합 네트워크 구조와 동일한 구조의 네트워크를 그림 5.9와 같이 구축하고 게이트웨이 어플리케이션, 모니터링 어플리케이션, 데이터베이스 어플리케이션을 연결하였다. 구축된 선박 통합 네트워크에 연결된 각 어플리케이션은 이 논문에서 제안한 미들웨어를 이용하여 IEC 61162-450을 만족하는 정보 교환 및 처리를 하였다. 또한 미들웨어를 구성하는 NF 모듈, SF 모듈, 서비스 모듈의 각 기능이 오류 없이 정상적으로 동작하는 것을 확인하였다.

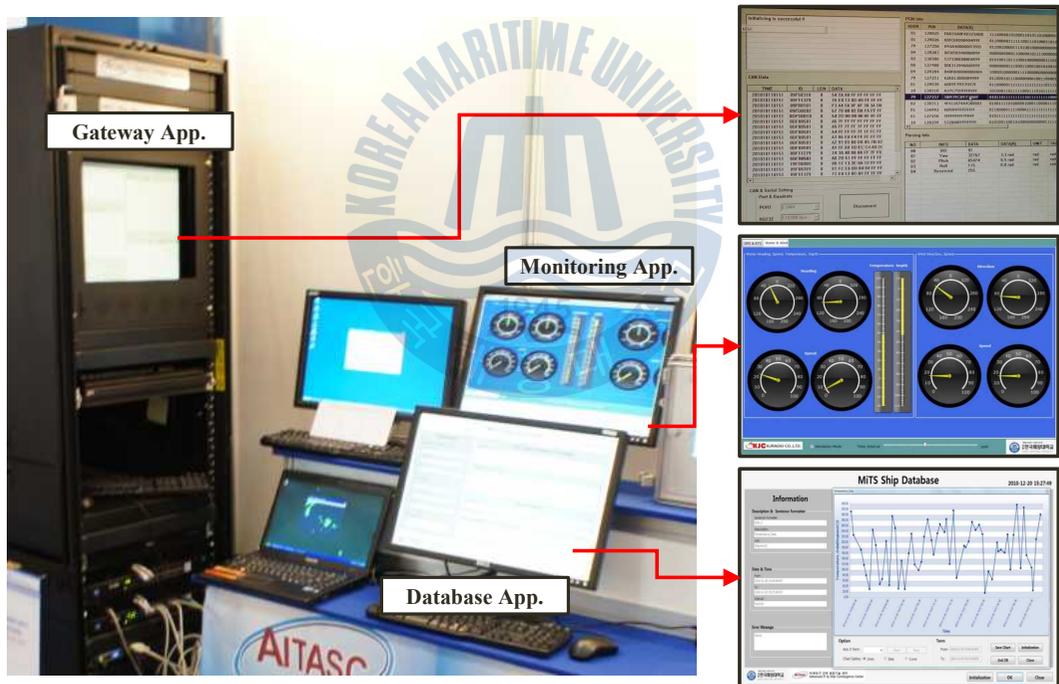


그림 5.9 시험을 위해 구축된 선박 통합 네트워크

Fig. 5.9 The shipboard integration network established for test

제 6 장 결론 및 향후 과제

기존의 선박 통합 네트워크의 표준인 IEC 61162-4 시리즈는 요구사항이 많고 시스템 구조가 복잡하여 구현이 어려운 문제점이 있었다. 이러한 문제점의 해결을 위해 IEC는 더 이상 IEC 61162-4 시리즈를 선박 통합 네트워크의 표준으로 유지하지 않기로 결정하였으며 이를 대체할 표준인 IEC 61162-450을 제정하고 있다.

이 논문에서는 대체될 표준인 IEC 61162-450을 분석하여 IEC 61162-450을 기반으로 하는 선박 통합 네트워크에서 시스템 간의 정보 교환 및 처리를 위한 미들웨어를 개발하였다. 개발한 미들웨어는 네트워크 기능(NF : Network Function block) 모듈, 시스템 기능(SF : System Function block) 모듈, 서비스 모듈로 구성되어 있다. NF는 선박 통합 네트워크 내에 있는 시스템 간의 상호 연결 및 데이터그램 송수신 등의 기능을 하며 SF는 송수신된 데이터그램의 메시지 처리, 유효성 검사, 오류 로그 등의 기능을 하고 서비스 모듈은 시스템이 NF와 SF의 기능을 이용할 수 있도록 인터페이스를 제공하는 기능을 한다. 개발한 미들웨어에 대한 시험을 통하여 IEC 61162-450에서 제시하는 성능 검사 및 성능 평가 요구사항을 만족하는 것을 검증하였고 선박 통합 네트워크를 구축하여 개발한 미들웨어의 유용성을 확인하였다.

개발한 미들웨어를 선박 통합 네트워크에 적용함으로써 선박 통합 네트워크 내에서 IEC 61162-450 표준을 만족하는 정보 교환 및 처리가 가능하며 선박 내에서 발생하는 정보 통합 관리의 편의성이 향상된다. 또한 이 연구를 통하여 IEC 61162-450 표준의 정식 발표에 신속하게 대처할 수 있다.

향후 선박 통합 네트워크의 표준인 IEC 61162-450이 정식 발표되면 추가적으로 요구하는 사항 및 수정 사항을 수용하여 이 논문에서 개발한 미들웨어의 확장에 관한 연구가 필요할 것이다.

참고문헌

- [1] 박휴찬, 이장세, 장길웅, 이정우, 정희섭, 박중현, 강순열, "선박에서의 통합 정보처리를 위한 시스템 아키텍처", 2009년도 한국마린엔지니어링학회 공동학술대회 논문집, pp. 207-208, 2009.
- [2] IEC, IEC61162-4 : Maritime Navigation and Radiocommunication Equipment and Systems - Digital Interfaces - Multiple Talkers and Multiple Listeners - Ship Systems Interconnection, 2001.
- [3] IEC, 61162-450 Ed.1 : Maritime Navigation and Radiocommunication Equipment and Systems - Digital Interfaces - Part 450: Multiple Talkers and Multiple Listeners - Light-weight Ship Systems Interconnection, 2010.
- [4] 이광일, 박준희, 최원석, 문경덕, "선내 통신 국제 표준화 동향", TTA Journal, 제126호, pp. 45-51, 2009.
- [5] 해양수산부, 국제해사기구(IMO) 제53차 항해안전전문위원회(NAV) 회의 결과, 2007.
- [6] 정중식, 김선영, "e-Navigation 서비스 제공을 위한 통신망 아키텍처의 설계", 한국항해항만학회지, 제32권, 제1호, pp. 37-45, 2008.
- [7] Lee A. Luft, Larry Anderson, and Frank Cassidy, "NMEA2000 A Digital Interface for the 21st Century", Institute of Navigation's 2002 National technical Meeting, 2002.
- [8] 유영호, "차세대IT선박 기술분석과 전망", 전자공학회지, 제35권, 제2호, pp. 19-29, 2008.
- [9] 박동현, 허진영, 김중현, 유영호, 심진보, 신옥근, "FPGA를 이용한 NMEA2000 선박 표준네트워크 프로토콜의 구현", 2009년도 한국마린엔지니어링학회 공동 학술대회 논문집, pp. 205-206, 2009.

- [10] 유영호, "NMEA와 NMEA 2000 통신프로토콜 표준화 동향", TTA Journal, 제 126호, pp. 52-57, 2009.
- [11] 이장세, 박휴찬, 장길웅, 이주형, 장남주, 이주영, 이부형, "선박 내 정보의 통합관리를 위한 정보 아키텍처", 2009년도 한국마린엔지니어링학회 공동학술대회 논문집, pp. 209-210, 2009.
- [12] 김태중, 황훈규, 서정민, 윤진식, 이장세, 장길웅, 박휴찬, "MiTS 통신을 위한 메시지 처리 모듈의 설계 및 구현", 한국해양정보통신학회논문지, 제14권, 제2호, pp. 360-368, 2010.
- [13] 황훈규, 김태중, 윤진식, 서정민, 박휴찬, 장길웅, 이장세, "선박 내 통합 정보 서비스를 위한 미들웨어 서버의 서비스 모듈 설계 및 구현", 한국마린엔지니어링학회지, 제34권, 제1호, pp. 141-146, 2010.
- [14] 정보통신부, 한국전자통신연구원, 침입감내기술 개발, 2005.
- [15] NMEA, NMEA 0183 : Standard for Interfacing Marine Electronic Devices Version 4.00, 2008.
- [16] Network Working Group, RFC 5424 : The Syslog Protocol, 2009.
- [17] 황훈규, 윤진식, 이성대, 서정민, 장길웅, 이장세, 박휴찬, "Light-Weight Ethernet 기반 MiTS 네트워크 프로토콜 개발", 한국마린엔지니어링학회지, 제34권, 제8호, pp. 164-171, 2010.
- [18] 황훈규, 윤진식, 서정민, 이성대, 이장세, 박휴찬, 장길웅, "Light-Weight Ethernet 기반 MiTS 서비스 모듈 개발", 한국마린엔지니어링학회지, 제34권, 제8호, pp. 172-179, 2010.