선박의 통합정보처리를 위한 IEC 61162-4 기반 TLI 프로토콜 설계 및 구현

장길웅* · 이장세** · 박휴찬**

Design and Implementation of IEC 61162-4 based TLI Protocol for e-Navigation on Ship

Kil-Woong Jang* · Jang-Se Lee** · Hyu-Chan Park**

이 논문은 2009년도 지식경제부 및 정보통신산업진흥원 IT핵심기술개발사업의 연구비를 지원받았음

요 약

본 논문에서는 선박의 통합정보처리를 위해 표준화된 IEC 61162-4 기반 TLI 프로토콜의 전송 알고리즘을 제안하고, 제안된 알고리즘을 기반으로 프로토콜을 설계 및 구현한다. 구현된 프로토콜은 TCP/IP 기반의 프로토콜로서 IEC 61162-4에서 제안하고 있는 MAU와 LNA의 표준에 따른 통신 프로토콜이며, 클라이언트/서버 모델을 기반으로 한 네트워크 구조에서 데이터 전송이 이루어진다. 개발된 TLI 프로토콜은 하나의 호스트 내에 MAU와 LNA간의 통신과 서로 다른 호스트의 LNA 간 통신이 이루어지며, 하나의 LNA에 대해 다양한 서비스를 제공하기 위해 멀티 MAU 구조로 구현된다. 표준 규격과 제안된 전송 알고리즘에 따라 객체지향 기법을 이용하여 프로토콜 설계 및 구현이 되었으며, 실제 선박에서 사용될 수 있는 간단한 네트워크 환경에서 실험을 하여 성공적으로 데이터 송수신이 됨을 확인할 수 있었다.

ABSTRACT

In this paper, we propose a transmission algorithm of the TLI protocol of the IEC 61162-4 standard to perform the e-Navigation on ship, and we design and implement the TLI protocol according to the proposed algorithm. The implemented protocol is a TCP/IP based protocol, and is a network protocol with the MAU and LNA components proposed in the IEC 61162-4 standard. In addition, it makes a data transmission over the network architecture based on client/server network model. In the implemented protocol, there are a communication between the MAU and the LNA in one host, and also a communication between the LNA and the LNA in each other hosts. In order to provide a variety of service in a host, every client host has a LNA and multiple MAUs. According to the standard specification and the proposed transmission algorithm, we designed and implemented the TLI protocol using object-oriented mechanism. We carried out the experiment under a simple network model similar to real ship environment, and confirmed that it successfully transmits and receives data between the hosts.

키워드

IEC 61162-4, 전송계층인터페이스, 선박, 통합정보처리

Key word

IEC 61162-4, TLI, ship, e-Navigation

* 한국해양대학교 데이터정보학과

** 한국해양대학교 IT공학부

접수일자 : 2009. 11. 20

심사완료일자 : 2009. 12. 04

I. 서 론

최근 조선산업은 선박 내에 장비 및 시스템들의 IT화로 인하여 고부가가치산업으로 발전하고 있다. 국내 조선산업은 막대한 투자와 기술집약 건조방식을 도입하여 선박건조 부문에서 세계 1위 수주량을 자랑하고 있으며, 국내 주력 수출산업으로 인정받으며 향후 지속적으로 발전할 것으로 예상된다[1]. 현재 선박의 장비와 시스템은 조선산업 및 조선기자재 산업의 발전과 더불어 발전하고 있지만, 선박내의 장비 및 시스템은 개별적으로 발전되어 왔다. 따라서 이러한 장비 및 시스템 간 통합 및 육상과의 정보공유 측면에서는 볼 때 초보단계 수준이며, 이것은 최근에 많이 발생되는 해양사고의 원인 중 하나로 지적되고 있다.

이러한 문제를 해결하고자 첨단 IT 기술 기반으로 선박 내 장비 및 시스템의 통합과 사용자 편의를 위한 통합항법장치의 국제 표준화가 시도되고 있으며, 이것은 e-Navigation이라는 개념으로 발전하고 있다. 여기서, e-Navigation이란 해상에서 안전, 보안 및 해양환경을 위해 운항 및 그와 관련된 서비스를 향상시킬 목적으로 전자적 방법을 사용하여 선박과 육상으로 해상정보에 대한 조화로운 수집, 통합, 교환, 표현 및 분석을 제공하는 것을 말한다[2, 3, 4, 5].

e-Navigation을 표준화하기 위해 국내에서도 많은 노력을 하고 있다. 구체적으로 표준화 대상항목을 정하여 학계 및 연구소, 기업에서 연구 중에 있으며, 대표적인 표준화 대상항목으로는 항법 장비 기술, 표준기자재 임베디드 시스템 기술, 교통정보 수집 기술 등이 있다. 이 중 항법 장비 기술은 MiTS(Maritime Information Technology Standard) 네트워크 기술, GIS/GPS 수신기술, 항해/항법 기술 등 첨단 IT 기술을 이용한 항법 기술을 포함한다.

본 논문에서는 선박 내 장치 및 시스템 간 통합을 이루기 위하여 MiTS 네트워크 기술 관련 표준인 IEC (International Electrotechnical Commission) 61162-4 표준안에서 제시하고 있는 프로토콜 구조를 분석하고, 효율적인 데이터 전송이 이루어질 수 있는 전송 알고리즘을 제안한다. 또한 표준안에서 제시한 프로토콜 구조 하에서 제안된 전송 알고리즘을 적용한 TLI(Transport Layer Interface) 프로토콜을 설계하고 객체지향기법을 사용하여 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 IEC 61162-4 표준에 대하여 기술하고, 3장에서는 제안된 IEC 61162-4 기반 TLI 전송 알고리즘에 대하여 기술한다. 4장에서는 표준과 제안된 알고리즘을 적용한 프로토콜 설계 및 구현과 관련된 내용을 기술하고, 마지막으로 5장에서 결론을 맺는다.

II. IEC 61162-4 통신 프로토콜

선박 내 시스템 간의 데이터 및 정보 교환을 위한 대표적인 통신 프로토콜은 MiTS이며, 이를 계승 발전한 것이 IEC 61162-4 표준안이다 [6]. IEC 61162-4 표준 통신 프로토콜은 그림 1과 같은 구조를 가진다. 상위 사용자 응용 프로그램과 하위 하드웨어 외에 크게 2 부분으로 나누어진다. 하나는 OSI 7 계층에서 상위 5에서 7 계층 기능을 가진 부분으로 A-Profile(Application Profile)이며, 다른 하나는 나머지 하위 계층 기능을 가진 부분으로 T-Profile(Transport Profile)이다.

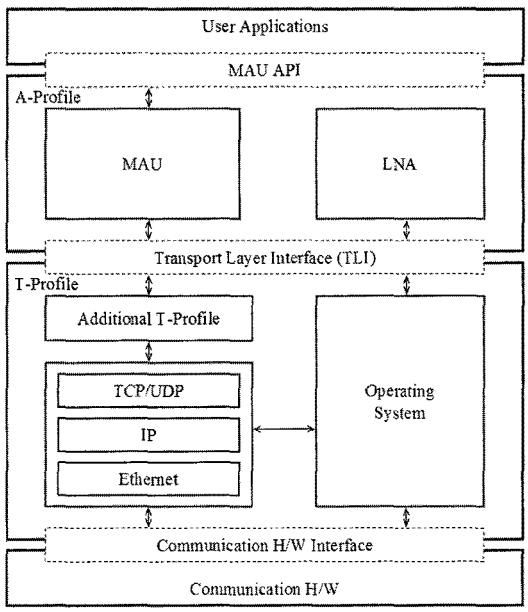


그림 1. IEC 61162-4 프로토콜 스택
Fig. 1 IEC 61162-4 protocol stack

A-Profile 내에는 MAU(MiTS Application Unit)와 LNA(Local Network Administrator) 컴포넌트를 가진다. MAU는 상위 응용 프로그램과 통신을 위한 기능을 담당하며, LNA는 하위 T-Profile과의 네트워크 인터페이스 역할을 담당한다. 표준에서는 한 호스트 내에 여러 개의 응용 프로그램을 처리하기 위해 멀티 MAU 구조와 멀티 MAU와 하나의 LNA가 TLI를 사용하여 통신이 가능한 구조를 제시하고 있다.

서로 다른 호스트 간에 통신은 그림 2와 같이 일반적으로 LNA와 LNA간의 통신으로 이루어진다. 하지만 특수한 경우에 LNA를 거치지 않고 MAU에서 직접 통신이 가능한 경우도 표준에서는 제시하고 있다. MAU/MAU 통신 또는 MAU/LNA 통신에서 TLI는 각 컴포넌트에 포함되어 각 컴포넌트 간의 데이터 통신 기능을 담당하고 있으며, 전송되는 데이터 종류에 따라 TCP 또는 UDP 방식을 사용하여 데이터를 전송한다.

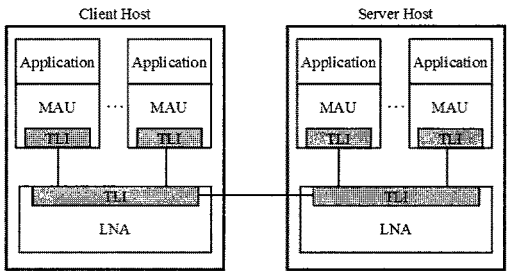


그림 2. MAU/LNA/TLI의 프로토콜 구조
Fig. 2 Protocol architecture of MAU/LNA/TLI

Ⅲ. IEC 61162-4 기반 TLI 전송 알고리즘

이 장에서는 IEC 61162-4 기반 TLI 프로토콜을 구현하기 위한 알고리즘을 기술한다. 프로토콜 설계를 위해 구현하고자 하는 시스템의 구조는 그림 3과 같다. 선박 내에는 네트워크로 연결된 수많은 장치와 시스템들이 탑재되어 있다. 그림에서 점선으로 이어진 네트워크는 NMEA 2000 [7]에서 동작하는 장치들 간의 네트워크를 나타내며, NMEA 2000 장비들은 이 네트워크를 통해 수집된 데이터들을 게이트웨이로 전송하게 된다. 전송된 데이터들은 게이트웨이에서 데이터가 필터링되며, 필터링된 데이터는 통신 프로토콜을 이용하여 서버로 전송되며, 전송된 데이터는 데이터베이스에 저장된다. 각 클라이언트는 필요에 따라 서버로 데이터 요청을 수행하게 된다. 이때 통신 프로토콜에 따라 요청이 이루어지며, 요청된 데이터는 서버에서 데이터베이스를 검색하여 각 클라이언트로 결과 데이터를 전송한다.

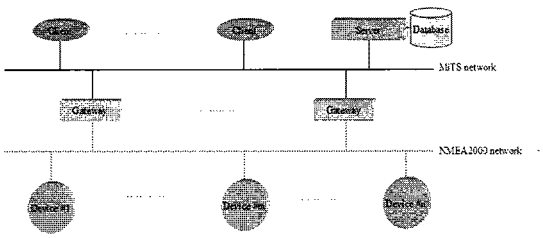


그림 3. 시스템 구조
Fig. 3 System structure

TLI 프로토콜은 클라이언트/서버 모델로 표현할 수 있다. TLI 프로토콜 동작을 간단히 설명하기 위해 하나의 클라이언트와 하나의 서버 상에서 일어나는 과정으로 기술한다. 상위 응용 프로그램에서 전송 요청이 오면 클라이언트 MAU에서는 클라이언트 LNA로 연결 요청 메시지를 전송하게 된다. 이때 서버의 주소를 담아 전송하게 된다. 메시지를 수신한 클라이언트 LNA는 응답 메시지를 전송하여 연결이 설정된다. 연결이 설정되면 클라이언트 MAU는 클라이언트 LNA로 데이터 요청 메시지를 전송하게 된다. 데이터를 수신한 클라이언트 LNA는 서버 LNA로 연결 설정을 요청한다. 연결 설정 메시지를 수신한 서버 LNA는 연결 설정 응답 메시지를 전송하여 두 호스트 간에 연결을 설정한다. 연결 설정이 이루어지면 서버 LNA로 데이터 요청 메시지를 전송한 후 대기한다. 데이터 요청을 수신한 서버 LNA는 서버 MAU로 연결 요청 메시지를 전송한 후 연결 응답 메시지를 수신한 후 데이터 요청 메시지를 전송한다. 서버 MAU는 연결이 설정되고 데이터 요청 메시지를 수신하면 요청된 데이터를 데이터베이스에서 검색하여 검색된 결과를 데이터 메시지에 담아 전송하게 된다. 전송되는 데이터 메시지는 이전 과정에서 설정된 연결을 따라 전송된다. 연결 기반 (connection-oriented) 방식

을 사용하기 때문에 한 번 연결 설정이 이루어진 이후에는 다시 연결 설정 없이 데이터를 전송하게 된다. 그림 4는 설계된 TLI 프로토콜의 시퀀스 다이어그램을 나타낸 것이다.

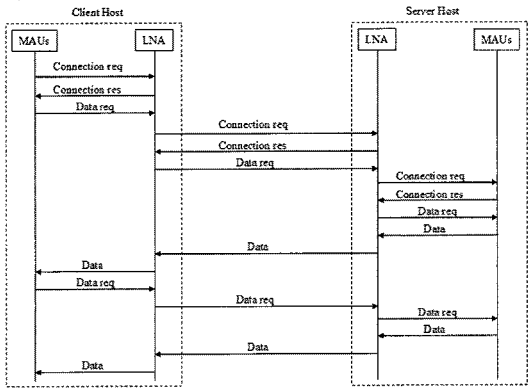


그림 4. TLI 프로토콜 시퀀스 다이어그램
Fig. 4 Sequence diagram of TLI protocol

다음은 클라이언트 및 서버의 각각의 MAU와 LNA에서 수행되어지는 전송 알고리즘에 대한 상세 설명을 기술한다.

3.1 로컬 MAU의 TLI 전송 알고리즘

클라이언트가 서버로 특정 데이터를 요청할 경우 우선 상위 응용 프로그램는 서버의 IP 주소를 포함한 데이터 요청 메시지를 클라이언트 MAU로 요청한다. 데이터 요청 메시지를 받은 클라이언트 MAU는 요청된 서버의 IP 주소로 이전에 연결요청이 이루어졌는지 검사한다. 만약 서버와 연결이 이루어지지 않았을 경우, 클라이언트 LNA로 TCP 연결 요청을 한다. 이때 클라이언트 LNA와는 루프백 IP 주소 (127.0.0.1)를 이용하여 연결한다. 클라이언트 LNA와 연결이 이루어지면 응용 프로그램에서 요청한 데이터 요청 메시지를 클라이언트 LNA로 전송하고 수신 대기상태를 유지한다. 클라이언트 LNA로 데이터 요청에 따른 결과 메시지를 수신하게 되면 그 결과를 응용 프로그램을 전송한다. 최초에 요청 메시지가 수신되었을 경우 클라이언트 LNA와 연결 요청이 먼저 이루어진 후 데이터 요청 메시지가 이루어지지만, 특정 서버와 연결 설정이 이루어진 이후에는 연결 요청 없이 설정된 연결을 통하여 메시지가 전송 및 수신된다. 그림 5는 클라이언트 MAU의 TLI 전송 알고리즘을 나타낸 것이다.

```
/* When the application requests data (D) from the server, check
whether a connection is or not between client MAU and client
LNA */
if (isConnectMAULNA(clientIP, serverIP)) {
  // send a data request message to client LNA
  requestData(D);
  // listen the results for the request from client LNA
  Result = listenData();
  // send the result to the application
  sendApp(Result);
}
else {
  // connection between client MAU and client LNA
  if (MAULNAConnect(clientIP, serverIP)) {
    requestData(D);
    Result = listenData();
    sendApp(Result);
  }
  else {
    // send a error message to the application
    sendApp(errMsg);
  }
}
```

그림 5. 클라이언트 MAU의 TLI 전송 알고리즘
Fig. 5 Transmission algorithm of TLI for client MAU

3.2 클라이언트 LNA의 TLI 전송 알고리즘

클라이언트의 MAU와 LNA간의 연결이 설정된 후, MAU로부터 데이터 요청 메시지가 수신되면, 클라이언트 LNA는 수신된 서버의 IP 주소를 이용하여 서버와 연결 설정이 되어있는지 검사한다. 만약 이전에 연결 설정이 이루어진 경우에는 데이터 요청 메시지를 설정된 연결로 전송한 후 대기 상태를 유지한다. 서버로부터 결과 데이터를 수신한 후에는 그 데이터를 이전에 클라이언트 MAU와 설정된 연결로 전송한다. 만약 이전에 서버의 LNA와 연결 설정이 이루어지지 않은 경우에는 우선 TCP 연결 설정 요청을 한 후 연결 설정이

이루어진 이후에 데이터 요청 메시지를 전송한다. 그림 6은 클라이언트 LNA의 TLI 전송 알고리즘을 나타낸 것이다.

```
/* After connecting between client MAU and client LNA, on
 receiving requests data (D) from client MAU, check whether a
 connection is or not between client LNA and server LNA */
if (isConnectLNALNA(clientIP, serverIP)) {
  // send a data request message to server LNA
  requestData(D);
  // listen the results for the request from server LNA

  Result = listenData();
  // send the result to client MAU
  sendMAU(Result);
}
else {
  // connection between client LNA and server LNA
  if (LNALNAConnect(clientIP, serverIP)) {
    requestData(D);
    Result = listenData();
    sendMAU(Result);
  }
  else {
    // send a error message to client MAU
    sendMAU(errMsg);
  }
}
```

그림 6. 클라이언트 LNA의 TLI 전송 알고리즘
Fig. 6 Transmission algorithm of TLI for client LNA

3.3 서버 LNA의 TLI 전송 알고리즘

클라이언트의 LNA와 서버의 LNA간에 연결 설정이 이루어진 이후에 클라이언트로부터 데이터 요청 메시지를 수신하면 서버 MAU와 이전에 연결 설정이 이루어졌는지 검사한다. 만약 연결 설정이 이루어져 있으면, 설정된 연결로 데이터 요청 메시지를 전송하고 수신 대기 상태를 유지한다. 서버 MAU로부터 결과 데이터를 수신하면 그 데이터를 이전에 클라이언트 LNA와 설정된 연결로 데이터를 전송한다. 만약 이전에 서버 MAU와 연결 설정이 이루어져 있지 않을 경우 서버 MAU와 TCP 연결을 설정한 후 데이터 요청 메시지를 전송하여 결과를 기다린다. 그림 7은 서버 LNA TLI 전송 알고리즘을 나타낸 것이다.

```
/* After connecting between client LNA and server LNA, on
 receiving requests data (D) from server LNA, check whether a
 connection is or not between server LNA and server MAU */
if (isConnectLNAMAU(clientIP, serverIP)) {
  // send a data request message to server MAU
  requestData(D);
  // listen the results for the request from server MAU
  Result = listenData();
  // send the result to client LNA
  sendLNA(Result);
}
else {
  if (LNAConnect(clientIP, serverIP)) {
    // connection between server MAU and server LNA
    requestData(D);
    Result = listenData();
    sendLNA(Result);
  }
  else {
    // send a error message to client LNA
    sendLNA(errMsg);
  }
}
```

그림 7. 서버 LNA의 TLI 전송 알고리즘
Fig. 7 Transmission algorithm of TLI for server LNA

3.4 서버 MAU의 TLI 전송 알고리즘

서버 LNA와 MAU가 TCP 연결 설정이 이루어진 후 데이터 요청 메시지를 수신한 서버 MAU는 서버 내의 데이터베이스에서 요청된 데이터를 검색한 후 결과를 이전에 설정된 TCP 연결로 그 결과 데이터를 전송한다. 그림 8은 서버 MAU의 TLI 전송 알고리즘을 나타낸 것이다.

```
/* After connecting between server MAU and server LNA, on
receiving requests data (D) from server LNA */
// receive a data request message from server LNA
D = receiveLNA();
// get the results of D from its database
Result = getData(D);
// send the results to server LNA
sendLNA(Result);
```

그림 8. 서버 LNA의 TLI 전송 알고리즘
Fig. 8 Transmission algorithm of TLI for server MAU

Ⅳ. IEC 61162-4 TLI 설계 및 구현

이번 장에서는 앞서 기술한 IEC 61162-4의 TLI 전송 알고리즘을 적용한 TLI 설계부분과 컴퓨터 언어를 이용한 구현부분을 기술한다.

4.1 TLI 설계

프로토콜을 구현하기에 앞서 효율적인 구현을 위한 프로토콜 설계가 필수적이다. 통신 프로토콜은 크게 전송부분과 수신부분으로 구분될 수 있다. IEC 61162-4의 표준은 크게 MAU와 LNA로 구분되고, MAU와 LNA는 모두 TLI를 포함한다. TLI는 내부적으로 전송부분과 수신부분을 포함하고 있으며, IEC 61162-4 표준에서도 전송부분과 수신부분을 제시하고 있다. 그림 9와 10은 표준에서 제시하고 있는 전송 및 수신 상태도를 효율적인 구현을 위하여 일부 수정한 그림이다. 상태도에서 화살표는 상태이동을 나타내며, 각 상태이동 화살표에 있는 텍스트는 수평선을 기준으로 위쪽은 이벤트를 아래쪽은 액션을 나타낸다. 예를 들어, 그림 9에서 최초 상태이동인 "TLICreate Port/Wait for connection request" 에서 "TLICreatePort"는 이벤트를 의미하며, "Wait for connection request"는 액션을 의미한다.

TLI 전송부분은 크게 TP_DEF, TP_REQ, TP_CON 상태로 구분된다. TP_DEF 상태는 연결요청을 위한 초기화 상태이며, TP_REQ는 연결요청이 발생될 경우 이를 처리하기 위한 상태이며, TP_CON은 상대 호스트와 연결 설정이 이루어진 후 데이터를 전송하는 상태를 나타낸다. TP_REQ와 TP_CON 상태에서는 연결 종료와 에러 이벤트가 발생할 수 있으며, 이러한 이벤트가 발생할 경우 프로그램이 종료가 되거나 TP_DEF 상태로 전이된다. IEC 61162-4 표준에서는 TP_ERROR 상태와 TP_CLOSE 상태를 따로 제시하고 있지만, 본 논문에서는 효율적인 구현을 위해 에러처리부분과 종료부분의 상태를 따로 두지 않고 수신 상태도와 비슷한 형태로 재구성하였다.

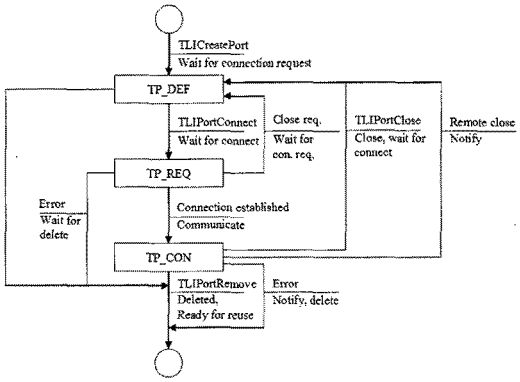


그림 9. TLI 전송 상태도
Fig. 9 Transmitter state diagram of TLI

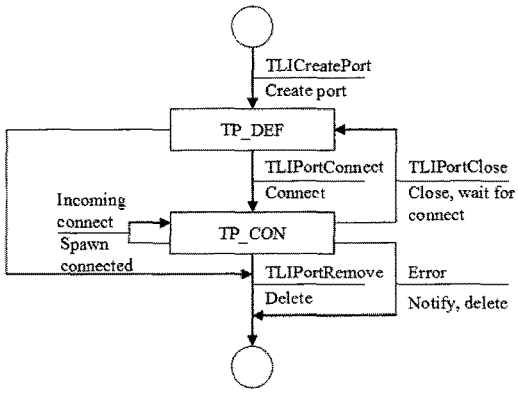


그림 10. TLI 수신 상태도
Fig. 10 Receiver state diagram of TLI

그림 10은 TLI의 수신부분의 상태도를 나타낸 것으로, TP_DEF과 TP_CON의 두 가지 상태가 있다. TP_DEF은 수신 초기화 상태이며, TP_CON은 연결설정 및 수신

대기 상태를 나타낸다. 전송부분과 마찬가지로 TP_CON 상태에서 연결 종료와 에러 이벤트가 발생할 수 있으며, 이때 프로그램이 종료가 되거나 TP_DEF 상태로 전이된다.

4.2 TLI 구현

본 논문에서는 제안된 전송 알고리즘과 프로토콜 설계 부분을 기반으로 프로토콜을 구현하였다. 효율적인 프로토콜 구현을 위하여 객체지향기법을 이용한 통신 프로토콜 구현기법[8]을 적용하였으며, 프로토콜 구현 환경은 Visual Studio 2008에서 객체지향 언어인 C++를 이용하였다. 그림 11과 12는 앞서 기술한 TLI의 전송부분과 수신부분에 해당되는 클래스 다이어그램을 나타낸 것이다. 클래스 구현은 크게 Context, State, Concrete State의 3부분으로 나누어진다.

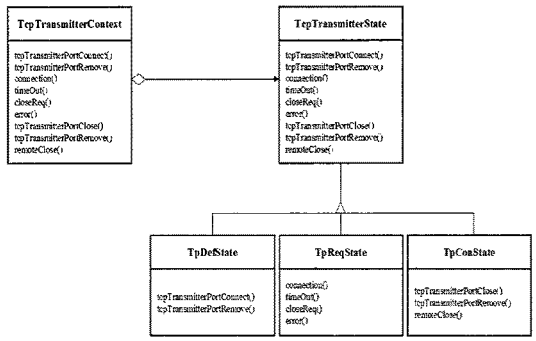


그림 11. TLI 전송부분의 클래스 다이어그램
Fig. 11 Class diagram for TLI transmitter

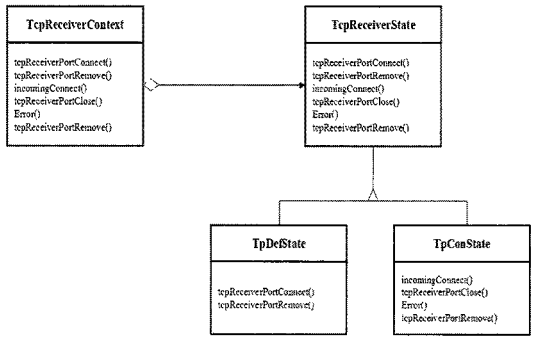


그림 12. TLI 수신부분의 클래스 다이어그램
Fig. 12 Class diagram for TLI receiver

Context 클래스는 프로토콜을 구성하는 이벤트 및 액션을 수행하는 메소드를 명시하고 있으며, State 클래스는 프로토콜 상태별로 메소드를 구성하는 가상 클래스이며, Concrete State 클래스는 State 클래스의 프로토콜 상태별로 Context 클래스의 각 메소드를 실제 구현하는 클래스이다.

그림 13은 본 논문에서 구현한 TLI 프로토콜을 이용하여 구성한 네트워크를 나타낸 것이다. 네트워크에서 호스트는 Gateway, APP, Server 3개가 있으며, 앞서 기술한 전송 알고리즘과 비교했을 때 Gateway와 APP는 클라이언트에 속하며, Server는 서버에 해당된다. Gateway와 APP는 멀티 MAU 프로세스를 가지며, 단일 LNA 프로세스에서 멀티 스레드를 이용하여 멀티 MAU를 처리하도록 구현하였다. 반면에 서버는 단일 MAU 프로세스와 단일 LNA 프로세스를 가지며, 각 클라이언트의 멀티 데이터 서비스를 처리하기 위해 MAU와 LNA는 멀티 스레드를 이용하여 구현하였다. 그림 13과 같은 네트워크 하에서 구현된 프로토콜을 이용하여 데이터 전송 실험을 하였다.

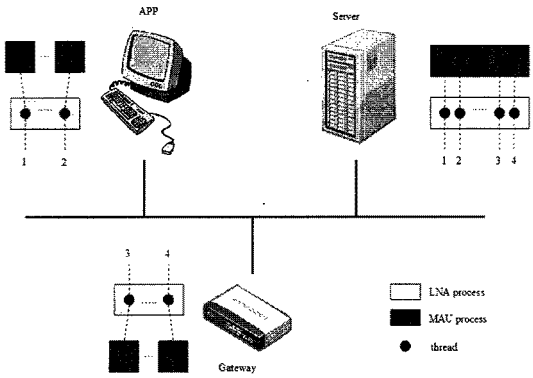
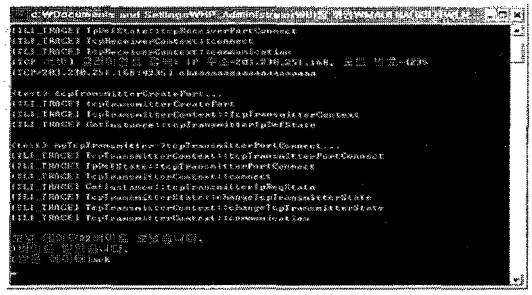


그림 13. 실험을 위한 네트워크 구성도
Fig. 13 Network structure for experiment

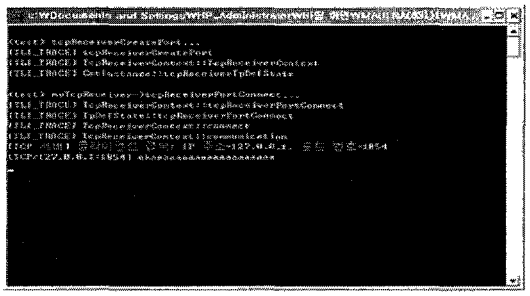
그림 14는 APP와 Gateway에서 다수의 응용 프로그램을 동작시켜 서버로 데이터 요청 메시지를 전송하고 그에 해당되는 결과를 전송하는 과정을 나타낸 것이다. 실험결과에서 제안된 전송 알고리즘과 표준에 따라 설계된 프로토콜이 전송 상태에 맞게 데이터 전송이 정상적으로 이루어짐을 볼 수 있었다.

(a)

(b)



(c)



(d)

그림 14. 구현된 TLI 프로토콜 실험 (a) 클라이언트 MAU (b) 클라이언트 LNA (c) 서버 LNA (d) 서버 MAU
Fig. 14 Experiment of implemented TLI protocol (a) Client MAU (b) Client LNA (c) Server LNA (d) Server MAU

V. 결 론

본 논문에서는 선박 통신 프로토콜 표준인 IEC 61162-4 기반 TLI 프로토콜의 전송 알고리즘을 제안하고, 제안된 알고리즘을 기반으로 프로토콜을 설계 및 구현하였다. 구현된 프로토콜은 IEC 61162-4 표준의 MAU와 LNA간 통신 기능을 담당하는 TLI 프로토콜을 표준 내용을 충실히 따르면서 TCP/IP 기반의 프로토콜로 설계 및 구현되었다. 프로토콜 구현을 위해 객체지향 기법을 이용하였으며, 간단한 네트워크 환경에서 실험한 결과 성공적으로 데이터 송수신이 됨을 확인할 수 있었다. 본 논문에서 제시한 알고리즘과 구현된 프로토콜은 다양한 장치들로부터 생성된 데이터를 통합적인 관리가 요구되는 e-Navigation에 효과적으로 적용될 수 있을 것으로 기대된다.

감사의 글

본 연구는 지식경제부 및 정보통신산업진흥원의 IT핵심기술개발사업의 일환으로 수행하였음. [2008-F-046-01, E-Navigation 대응 IT-선박 융합 핵심기술 개발]

참고문헌

[1] 임동선, 김재명, 박운용, "조선-IT 융합기술 e-Navigation 동향", TTA Journal, No. 119, 2008.

[2] 권혁동, "항로표지 지능화 E-Navigation 발전 전략", 해양한국(월간해양한국), 2007.

[3] 서기열, 서상현, "차세대 해상항법체계(e-Navigation)의 구현 방향", 전자공학회지 제34권, 제11호, 2007.

[4] http://site.ialathree.org/pages/FAQse-nav.pdf, e-Navigation Frequently Asked Questions.

[5] IMO, Maritime Safety Committee 81st Session, Agenda Item 23, Development of an E-Navigation Strategy, 2005.

[6] IEC 61162-4: Maritime Navigation and Radiocommunication Equipment and Systems - Digital Interfaces - Multiple Talkers and Multiple Listeners - Ship Systems

Interconnection, 2001.
[7] NMEA 2000: Standard for Serial-Data Networking of Marine Electronic Devices, 2004.
[8] H. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design Patterns: Elements of Reusable Objected-Oriented Software", Addison-Wesley, 1994.

저자소개

장길웅(Kil-Woong Jang)

1997년 경북대학교 컴퓨터공학과 (공학사)
1999년 경북대학교 컴퓨터공학과(공학석사)
2002년 경북대학교 컴퓨터공학과(공학박사)
2003년~현재 한국해양대학교 데이터정보학과 (부교수)
※관심분야: 네트워크 프로토콜, 유비쿼터스 네트워킹

이장세(Jang-Se Lee)

1997년 한국항공대학교 컴퓨터공학과(공학사)
1999년 한국항공대학교 컴퓨터공학과(공학석사)
2003년 한국항공대학교 컴퓨터공학과(공학박사)
2004년 ~ 현재 한국해양대학교 IT공학부(조교수)
※관심분야: 컴퓨터보안, 지능시스템, 모델링 및 시뮬레이션

박휴찬(Hyu-Chan Park)

1985년 서울대학교 전자공학과(공학사)
1987년 한국과학기술원 전기및 전자공학과(공학석사)
1995년 한국과학기술원 전기및전자공학과(공학박사)
1997년 ~ 현재 한국해양대학교 IT공학부(교수)
※관심분야: 데이터베이스, 데이터마이닝, 해양정보 시스템