

효율적인 IVEF 서비스를 위한 모바일 VTS 미들웨어 설계 및 구현

박 남 제[°]

Design and Implementation of Mobile VTS Middleware for Efficient IVEF Service

Namje Park[°]

요 약

IVEF 서비스는 해상교통관제시스템 상호간, 선박 상호간에 해상교통정보를 교환하기 위한 표준안으로서 e-Navigation이라 불리는 차세대 선박항법체계의 일환으로 표준화가 진행 중에 있다. 국제항로표지협회 IALA는 V-145 권고안을 통해 IVEF 서비스 모델과 서비스 제공을 위한 프로토콜을 제시하고 있으나 구체적인 구성은 이를 이용하는 사용자가 설계하여야 한다. 본 논문에서는 V-145 권고안의 기본 서비스 모델과 프로토콜을 토대로 해상교통정보의 교환을 편리하게 해주는 J-VTS 미들웨어를 설계하고 구현하였다. J-VTS 미들웨어는 IVEF 서비스를 제공하기 위한 다양한 컴포넌트들과 IVEF 메시지 프로토콜을 처리하기 위한 컴포넌트들로 구성된다. 상위 애플리케이션에 해당하는 해상교통관제시스템과 선박들은 J-VTS 미들웨어에서 제공하는 함수들만을 이용하여 편리하게 IVEF 서비스를 이용할 수 있도록 설계 되었으며, 접속자의 보안등급에 따라 서비스가 단계적으로 제공되게 된다.

Key Words : IVEF, VTS, e-Navigation, Mobile Middleware, Ocean Information Communication

ABSTRACT

The IVEF service is the draft standard designed for exchange of information on sea traffic between the vessel traffic systems and between the vessels. Standardization of this service is under way as a part of the next-generation navigation system, called e-Navigation. The International Association of Lighthouse Authorities (IALA) suggests, on its recommendation V-145, the IVEF service model and the protocol for provisioning of this service. But the detailed configuration of this service must be designed by the users. This study suggests, based on the basic service model and protocol provided in the recommendation V-145, the implementation of the J-VTS middleware which will facilitate exchange of information on sea traffic. The J-VTS middleware consists of various components for providing the IVEF service and for processing the IVEF message protocols. The vessel traffic systems and the vessels corresponding to upper-layer applications may use the IVEF service with the functions provided by the J-VTS middleware, and the services are designed to be accessed according to the security level of users.

※ 이 논문은 2013년도 정부(교육부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(과제번호:2013R1A1A4A01013587)

•° First Author and Corresponding Author : Department of Computer Education, Teachers College, Jeju National University, namjepark@jejunu.ac.kr, 정회원

논문번호 : KICS2013-11-520, Received November 30, 2013; Revised March 5, 2014; Accepted April 15, 2014

I. 서 론

2006년 국제해사기구 IMO(International Maritime Organization)에서는 해사안전위원회인 MSC(Maritime Safety Committee) 81차 회의에서 e-Navigation에 대한 논의를 시작하였다^{1,4)}. 2008년 MSC 85차 회의에서 e-Navigation의 전략을 승인하여 2012년까지 국제항로표지협회인 IALA(International Association of Lighthouse Authorities)를 중심으로 e-Navigation의 표준화 및 실용을 추진하고 있다^{1,3)}. e-Navigation은 해상활동에서의 안전 및 보안 또는 해양환경 보호를 목적으로 IALA를 중심으로 기술의 표준화 및 이행전략을 수행하고 있으며, 현재까지도 연구가 활발히 진행되고 있다⁴⁾. e-Navigation에는 VTS, AIS, ECDIS, IBS/INS, ARPA, LRIT, GMDSS 이렇게 7가지 분야로 구성되어 있다^{2,8)}. VTS(Vessel Traffic System)는 해상교통관제시스템을 의미하며 지금까지 국제해사기구(IMO)에서는 VTS간 정보교환에 대한 국제표준화를 발표한적 없다. IALA에서 개발한 VTS간 프로토콜 중 IVEF(Inter-system VTS Exchange Format)가 있다⁴⁾. IVEF는 VTS와 VTS간의 정보를 교환할 수 있도록 개발한 프로토콜로서 국제 표준화가 진행 중이다. 그러나 국내의 IVEF 기술 분석 및 도입에 관한 연구 및 계획은 거의 전무하다.

우리나라에서도 IVEF 기술 분석을 통하여 국내 해양 산업에도 국제 표준을 도입할 준비 및 기술 보안을 수행하도록 해야 한다. 본 논문에서는 오픈소스인 IVEF SDK(Software Development Kit)를 개선하여 무선 환경 하에서의 IVEF 서비스를 시뮬레이션 할 수 있는 시스템을 구현하였다¹⁴⁾. 항해 중인 선박을 에뮬레이션 하기위해 안드로이드 플랫폼이 탑재된 모바일 폰을 이용하였고, VTS 센터를 에뮬레이션 하기 위해 데스크톱 컴퓨터를 이용하였다. 본 논문에서는 VTS 미들웨어 구성요소 중 XML 파서에 관한 간략한 이론적 내용을 살펴보고, 파서의 실제구현에 대해 설명한다. 그리고 이들 파서들을 이용하여 구현된 V-145 프로토콜의 성능비교 결과를 비교하여 살펴본다. 2장에서는 IALA의 IVEF 서비스 프로토콜에 대해 설명하고, 3장에서는 본 논문에서 구현한 시스템의 구조 및 기능에 대해 설명하며, 마지막으로 결론을 맺는다.

II. 관련 연구

2.1 IVEF 서비스

IVEF 서비스는 2008년 IALA-AISM의 e-Na-

vigation working group에 의해 개발된 선박 교통 정보 교환 프레임워크이다. IALA를 통하여 국제적으로 데이터 공유를 위한 구조에 대한 연구와 표준 포맷이 정해지고 있으며 IVEF는 이중의 한 중요한 요소이다. 이러한 실시간의 트래픽 정보는 국가적으로 민감한 요소이나 현 규격에 정해진 표준 프로토콜이 미비하여 지속적인 발전이 요구되고 있다. IVEF 서비스는 VTS 시스템 상호간에 교통상황정보를 교환하기 위한 프로토콜로서 서버/클라이언트 모델로 동작한다. 오픈소스 기반으로 IALA에서 개발이 진행 중이며, OpenIVEF 사이트³⁾를 통해 SDK를 다운받아 프로토콜과 예제 프로그램을 확인해 볼 수 있다^{1,3)}. 서버/클라이언트 간에 서비스 제공을 위한 기본 동작은 그림 1과 같이 3단계로 이루어진다. 첫 단계로, 클라이언트는 서버에 인증을 요청하고 정당한 사용자인 경우 확인된 인증 응답을 받게 된다. 다음 단계로 서버는 해당 사용자를 위한 특정한 서비스가 있을 경우 이를 제공하게 되며, 없을 경우에는 BIS(Basic IVEF Services)라고 불리는 표준에서 정의한 기본적인 서비스를 제공하게 된다. 이때 클라이언트는 선호에 따라 관심영역, 데이터의 갱신 주기, 데이터의 형태 등을 지정할 수 있다. 마지막 단계로 클라이언트는 IVEF 서비스 사용의 종료를 위해 로그아웃 메시지를 서버에 보내게 된다. 로그아웃 메시지는 서버로부터 별도의 응답이 없으므로 클라이언트는 메시지 전송과 동시에 서버로의 접속을 해제하면 된다⁴⁾.

IALA는 VTS 센터 간에 IVEF 서비스를 제공하기 위한 기본적인 프로토콜로서 9개의 메시지를 정의하고 있다. 이들 메시지들의 정의는 XML 형태의 스키마로 구성이 되며, 모든 메시지들은 최상의 요소인 MSG_IVEF의 하위 요소로 이루어지게 되며, 각각의 하위요소 메시지들 또한 메시지들의 특성에 따라 별도의 하위 요소들을 가지고 있다. IVEF 메시지들은 크게 제어정보 메시지와 실시간 정보제공 메시지로 구분되어지며, 상호연결되어 있는 구조로 구성된다.

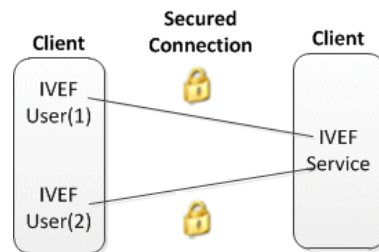


그림 1. IVEF 서비스 기본 개념도
Fig. 1. Conceptual Diagram of IVEF Service

제어정보 메시지는 사용자 인증과 종료, 서버로의 서비스 요청과 이에 대한 응답 메시지, 서버의 상태 정보 제공을 위한 메시지들로 구성된다. 실시간 정보제공 메시지는 선박의 현재 위치, 예상 항해경로, 정착지 및 기타 선박의 물리적 정보에 관련된 각종 사항들을 객체데이터(Object Data)로 관리한다. 객체데이터는 Real-time Tracking Positions, Static Vessel Information, Voyage related Information와 같은 정보의 교환을 주목적으로 한다^{5,9)}.

2.2 IVEF 서비스 모델

IVEF 서비스 정의는 'IALA Recommendation on Generic e-Navigation Service Engineering Model Template'에 따라 정의된다^{3,4)}. 당해 문서는 서비스의 전체 구성에 관한 요소들에 관한 정확한 세부 사항을 포함하고 있다. 구성요소 중 Basic Information Services는 IVEF 서비스를 제공하는 시스템이 기본적으로 구비해야 하는 서비스로서 제공하는 IVEF 서비스들에는 어떠한 것이 있으면 이에 대한 인터페이스는 무엇인지에 관한 기술 사항으로 구성되어 있다. Basic IVEF Services의 주요 컴포넌트는 데이터 모델(Data Model), 인터랙션 모델(Interfacing Model), 보안 모델(Security Model), 품질인자(Quality Parameters), 테스트 모델(Test Model), 관리자 모델(Administration Model)의 7개로 구성된다^{3,4)}. 데이터 모델은 IVEF 서비스의 데이터 타입과 오브젝트들을 정의하는 모델이다. IVEF 서비스에 의해 제공되는 데이터들의 추상적인 서술로서 인근해 e-Navigation 시스템과 외부 시스템 사이에 주고받는 데이터에 대한 사항이 정의된다. 인터랙션 모델은 IVEF Service의 개별 컴포넌트에 접근하기 위한 데이터 플로우(Data flow)와 프로토콜을 정의한다. 인터페이스 모델은 IVEF 서비스와 클라이언트간, 서비스로부터 요청하는 데이터들 간에 데이터 암호화와 인터페이스 프로토콜을 정의하는 모델이다. 보안 모델은 IVEF 서비스를 제어하기 위한 모델로서 접근 권한과 필터링과 같은 기능들을 정의하고 있다. 품질인자는 IVEF Service의 신뢰성과 성능을 조절하는 파라미터들에 관한 모델이다. 테스트 모델은 IVEF 서비스의 테스트와 유효성을 검사하는 기능을 한다. 관리자 모델은 IVEF 서비스의 기능적인 관리와 유지를 위한 관리자 기능에 관해 정의한다.

IVEF 서비스의 데이터 모델은 IALA Universal Maritime Data Model의 일부분이 된다. 이는 IVEF Service를 통하여 교환되는 데이터에 관해 잘 정의된

추상화 모델을 제공한다. 데이터 모델 자체는 서비스를 통해 송수신 되는 데이터의 암호화에 대한 사항은 포함하고 있지 않다. 데이터 모델과 보안 모델을 분리한 것은 주무관청에 의해 상황에 적합하다고 생각되는 보안 기술의 선택을 자유롭게 하기 위해서이다. 응용의 예를 들면, VTS 무선 통신 레이다, CCTV(Closed-circuit television)를 통해 인근해의 선박정보를 수집하게 되고 IVEF Service를 통해 다른 VTS와 선박정보를 주고받게 된다. VTS 센터는 선박정보를 필요로 하는 각종 응용 프로그램 장치들과도 IVEF 서비스를 이용하여 데이터를 송수신 한다. IVEF 서비스는 클라이언트/서버 모델로 동작하게 되며, 인근해에 있는 e-Nav igation 시스템의 일부로써 기능들이 서비스 시스템으로 연동되어 요청서비스들을 수행하게 된다.

2.3 종류별 XML 파서 플랫폼

XML 파서란 이러한 XML의 문서 구조 접근을 위한 XML의 관련된 기술로서 DOM, SAX, 안드รอย드 플랫폼에서는 SAX와 비슷하게 동작하는 XmlPullParser가 있다⁶⁾. DOM 파서는 프로그램이나 스크립트에서 동적으로 문서의 내용, 구조, 형식을 액세스하고 변경할 수 있게 하는 플랫폼과 언어 중립적인 객체모델 기반의 API로 노드(Node)단위로 데이터를 처리한다. 문서의 구조정보와 내용을 객체로 이용하며 메모리 적재방식이다. DOM 파서는 문서를 수정, 삭제, 추가하고자 할 때, 랜덤 액세스를 하고자 할 때, 새로운 데이터 구조를 생성하고자 할 때와 같은 상황에 적용이 가능하다⁶⁾.

SAX 파서는 XML을 사용하는 웹 파일들을 해석할 수 있게 해주는 이벤트 중심의 응용프로그램 인터페이스이다. 프로그래머가 일어날 수 있는 이벤트를 설정해 놓으면, SAX 파서는 그 이벤트가 일어났을 때 제어권을 가지고 상황을 처리한다. 이벤트 기반 API로서 XML 태그 시작과 끝에서 이벤트를 생성한다. XML 문서를 하나의 긴 문자열로 간주하며 라인 인터프리터방식으로 DOM 파서에 비해 단순한 인터페이스이나, 데이터 내용을 조작할 수 있는 기능은 상대적으로 적다. SAX 파서는 큰 문서를 핸들링 하고자 할 때나 빠른 처리를 요할 때(단, 부대조건을 따진 후), 이벤트 스트림에서 문서를 필터링하고자 할 때, 데이터 일부만이 필요할 때, 문서 자체에 대한 변경이 필요하지 않을 때와 같은 상황에 적용이 가능하다⁶⁾.

XmlPullParser는 XMLPULL V1 API에서 제공되는 인터페이스이다. 속도는 SAX보다 약간 느리며 메

과 동일한 동작을 하게 구현하였다.

3.1 IVEF 클라이언트 구현 및 주요 기능

안드로이드(Android)는 구글에서 개발한 플랫폼으로서 휴대 전화를 비롯한 휴대용 장치를 위한 운영 체제와 미들웨어, 사용자 인터페이스 그리고 표준 응용 프로그램을 포함하고 있는 소프트웨어 스택이자 모바일 운영 체제이다^[6,18]. 최근에 출시되는 안드로이드 플랫폼이 탑재된 스마트 폰의 경우 GPS(Global Positioning System) 센서가 기본으로 탑재 있어 안드로이드 SDK에서 제공하는 API를 이용하여 손쉽게 현재의 위치정보를 확인할 수 있다. 또한 구글 맵을 이용하여 해도를 에플리케이션을 할 수 있고, 기존 자바에서 사용하던 TCP/IP 프로토콜 스택을 이용할 수 있어 네트워크 프로그램을 작성하는데도 용이하다^[7,15]. 따라서 고가의 선박자동식별시스템(AIS, Automatic identification system)이나 전자해도정보시스템(ECDIS, Electronic Chart Display and Information System)가 없어도 특정 장치들을 에플리케이션 하기에 용이하고, 무선 통신을 통해 실제 선박과 VTS 간의 데이터 통신을 시뮬레이션 할 수 있다.

IVEF 클라이언트 프로그램은 안드로이드 버전 2.3.3(진저브레드)^[7]을 기반으로 작성되었으며, 2.3.3 이상 버전의 안드로이드 플랫폼 상에서 동작되게 구현하였다. 최신 버전인 4.1 젤리빈 환경 상에서도 정상 동작 유무를 확인하였다. 클라이언트 프로그램은 IALA의 V-145 권고 안에 따라 표준형 9개 주요 메시지 프로토콜들을 구현하였으며, 초기 구동 시에 그림 4에서와 같은 시작화면이 잠시 보인 후에 V-145 권고안의 메시지들을 테스트 할 수 있는 리스트 뷰가 표시된다. 이 후 사용자가 리스트 뷰의 특정 아이템을 선택하면 선택된 메시지에 해당하는 IVEF XML 메시지를 구성하여 서버에 요청을 수행한다. 서버는 이에 해당하는 서비스를 구동하고 관련 정보를 IVEF 메시

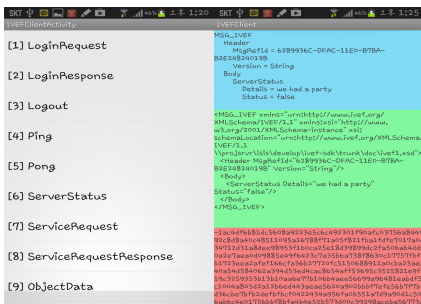


그림 4. IVEF 클라이언트 프로그램 실행화면
Fig. 4. IVEF Client program's Screen

지로 구성하여 클라이언트 단에 송신하게 된다. 클라이언트에 수신된 메시지는 그림 4의 화면에서와 같이 ① IVEF XML 메시지를 분석해서 획득한 주요정보와 ② 서버 단으로부터 전송받은 XML 메시지 ③ AES 알고리즘으로 암호화 된 보안된 형태의 문자열로 표시된다. AES 알고리즘은 64비트의 비밀키를 이용하여 암호화/복호화가 수행되며 본 개발에서는 EIVEF_Security 클래스 안에 비밀키가 “7de8e342f7cbac93”로 정의 되어있다.

그림 5는 옵션 메뉴와 서버 설정화면 및 Wi-Fi 연결이 되지 않았을 때 보여 지는 경고 다이얼로그 메시지이다. 클라이언트가 서버로 접속하기 위해서는 IVEF 서버 프로그램 실행 시 Server Information 창에 보여 지는 서버 IP와 서버 포트를 클라이언트의 서버 셋팅 창에 입력하여야 한다. 서버 설정 메뉴는 V-145 프로토콜 리스트 뷰 화면에서 옵션 메뉴를 선택하면 나오는 하위 메뉴 중 Server Setting 메뉴를 선택하여 설정할 수 있다. 이때 “Local Test On” 메뉴는 서버를 거치지 않고 폰 상에 저장되어 있는 파일정보를 이용하여 서버에서의 동작을 에플리케이션 할 수 있는 설정이다. 무선통신을 테스트하기 위해서는 3G 망이 아닌 Wi-Fi 를 통해서만 이용이 가능하다. 사용자가 3G 망으로 접속하는 것을 방지하기 위하여 Wi-Fi 설정이 되지 않은 상태에서는 그림 5의 3번째 화면에서와 같이 WiFi 설정을 확인 하라는 다이얼로그가 보이게 된다. 서버의 IP와 Port 그리고 폰 상의 WiFi 가 제대로 설정되면, 이 후부터 그림 5의 2번째 화면의 리스트 뷰에서 보이는 9가지의 V-145 권고안에 따른 메시지 요청을 서버로 전송하고 이에 해당하는 응답 메시지를 볼 수 있다.

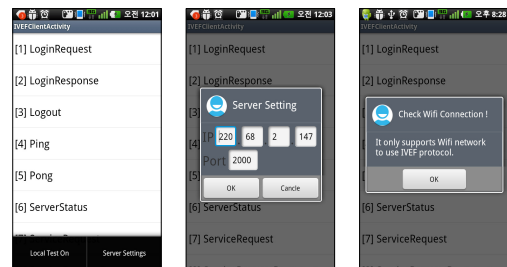


그림 5. Option 메뉴/서버설정, WIFI 연결 화면
Fig. 5. Option Menu/Server Set-up, WIFI-connect

3.2 IVEF 클라이언트 콘솔 프로그램 구현

IVEFClientConsole 프로그램을 실행하기 위해서는 실행폴더에 있는 run.bat을 실행하면 된다. 실행 화면은 그림 6과 같다. 서버의 IP와 Port를 콘솔(Console)

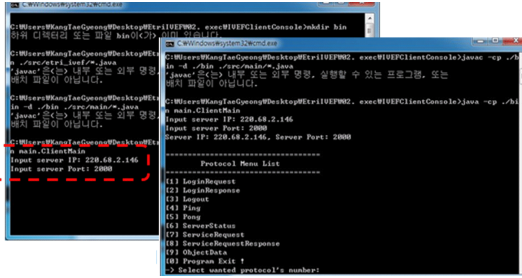


그림 6. IVEF 클라이언트 콘솔 프로그램 실행 화면
Fig. 6. Screen of IVEF Client Console Program

의 입력란에 적어주고, 이후부터는 콘솔 상에서 보이는 메뉴에서 테스트를 원하는 프로토콜을 선택하여 명령을 수행하면 된다. IVEFClientConsole 프로그램은 사용자가 선택한 메시지를 서버에 요청하고 EtriIVEF library를 이용하여 응답받은 수신 정보를 파싱하여 화면에 디스플레이하는 기능을 한다

3.3 IVEF 서버 서비스 시스템 구현

서버 프로그램의 주된 기능은 특정포트를 통해 클라이언트로부터 요청되는 V-145 권고 메시지를 수신하고, 요청받은 메시지를 분석하여 적절한 항해정보를 제공하는 것이다. V-145 권고안에 따라 BIS 서비스는 기본적으로 구현되어 있어야 하며, 부가적으로 별도의 서비스를 제공할 수도 있다. 서버 프로그램의 동작 순서는 그림 7과 같다. 처음 서버를 실행 시키면 클라이언트의 접속을 대기하는 상태로 진입한다. 클라이언트의 접속 요청이 있으면 서버는 클라이언트를 접속을 받아들이고 클라이언트로부터 데이터 문자열을 수신하게 된다. 클라이언트의 접속 과정에서 보안 모델을 적용하여 사용자 인증과정이 추가될 수도 있다^{12-14,19}. 서버로 송신된 데이터 문자열은 보안을 위해 64비트의 비밀키를 이용한 AES(Advanced Encryption Standard)

알고리즘으로 암호화 되어져 있다. 따라서 복호화를 거쳐야 온전한 형태의 IVEF 메시지가 생성된다. 본 개발에서는 비밀키를 서버와 클라이언트에서 이미 알고 있다는 가정하에 AES 알고리즘을 적용하였다. 복호화를 거친 후 IVEF 메시지는 V-145의 권고안에 따라 구성되어 있는지 유효성 체크가 이루어진다. XML 파일 형태 수준에서 적절한 XML 문법에 따라 작성되어져 있는지의 검증이 먼저 이루어지고, 다음으로 V-145 권고안에 따라 XML 로직이 구성되어 있는지 검증하게 된다. 유효한 메시지로 판별되면 DOM(Document Object Model for XML) 파서로 구현된 프로토콜 분석기(Protocol Analyzer)에서 테이블 1의 IVEF 인터페이스 메시지 중 어느 메시지인지를 판별하게 된다. 이때 IVEF 메시지 중에서 유용한 정보들이 데이터 컨테이너에 저장되며, 이 정보가 서버의 상태 정보 창에 갱신된다. 메시지의 분석이 끝나면 서비스 제공자(Service Provider)는 해당하는 메시지의 응답에 필요한 서비스를 구성한다. 현재 프로젝트의 개발에서는 소스 리팩토링의 구현이 주 목적이라 서버의 기능은 단순히 루프백(Loopback)만 하는 형태로 구현되었다. 마지막으로 서비스제공자는 V-145 권고안에 따라 XML 메시지를 생성하고, AES 알고리즘을 통한 암호화를 거쳐 클라이언트에 전송한다.

서버 프로그램의 화면 구성은 그림 8과 같이 서버 구동 시작과 멈춤을 위한 “Server Connection” 메뉴와 프로그램 종료를 위한 “Program Exit” 메뉴 그리고 6개의 상태창과 1개로 로그 이미지로 구성되어져 있다. 서버가 클라이언트의 접속 대기 상태에 들어가기 위해서는 그림 8의 왼쪽 화면과 같이 “Server Connection”의 “Server Start” 메뉴를 선택해 주어야 한다. 선택이후 서버는 클라이언트의 접속 요청을 받을 수 있게 되고, 클라이언트가 요청한 IVEF 메시지를 분석하여 관련 정보들이 계속적으로 6개의 상태창에 표시되게 된다. 6개의 상태창은 “Server Information”, “Server Status Monitoring”, “Useful Information”, “Request Protocol”, “Sending XML Message”, “AES Encoded XML message” 등이다. “Server Information” 창은 현재 서버가 구동되고 있는 PC의 서버 IP와 서버 Port 정보, 컴퓨터 이름을 보여주며 이 정보들을 이용하여 client의 서버 접속이 이루어진다. “Server Status Monitoring” 창은 현재 구동되고 있는 서버의 각종 상태를 표시한다. 현재 서버가 클라이언트의 대기 상태에 있는지 클라이언트 메시지를 처리하고 있는지 표시하며, 접속된 클라이언트의 IP 정보와 접속시간, 클라이언트 PC의 이름 등을

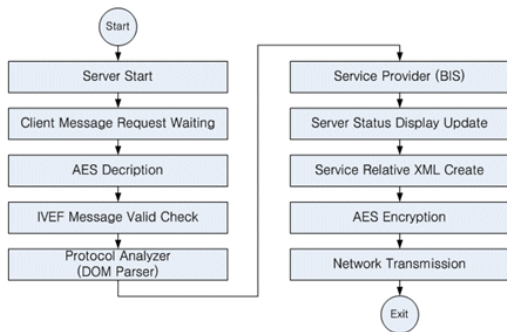


그림 7. IVEF 서버 서비스 동작 순서도
Fig. 7. Flow chart of IVEF Server operation

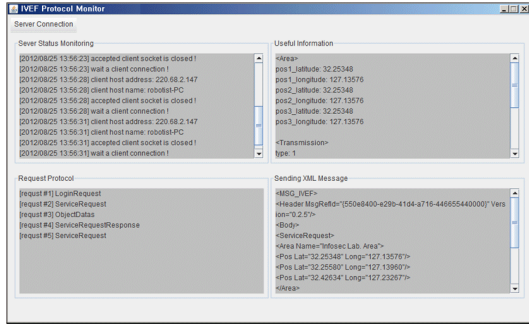


그림 8. IVEF 서버 실행화면
Fig. 8. Operation Screen of IVEF Server

보여준다. “Request Protocol”은 프로토콜 분석기를 통해 분석된 클라이언트의 요청 메시지를 순차적으로 보여 준다. 서버는 분석된 요청 메시지에 따라 필요한 서비스를 클라이언트에 제공해야 한다. “Useful Information” 창은 프로토콜 분석기를 통해 분석된 IVEF 메시지의 요소들을 사용자들이 보기 편하게 변환하여 상태창에 보여준다. 이때 IVEF 메시지의 각종 요소들은 메시지 특성에 따른 컨테이너 구조체로 데이터 정보가 관리된다. “Sending XML Message” 창은 클라이언트에 전송될 메시지들이 V-145 권고안에 따라 XML 형태의 포맷으로 변환된 텍스트를 보여준다. “AES Encoded XML message” 창은 서버가 전송할 XML 문서를 64비트의 비밀키를 이용하여 AES 암호화 시킨 16진수 문자열을 보여준다. 64비트의 비밀키는 서버와 클라이언트가 동일한 값을 알고 있어야 한다. 이렇게 암호화 된 문자열은 TCP/IP 스택을 통해 클라이언트에 전송되며, 전송이 종료되면 서버는 클라이언트와의 접속을 끊고 다시 접속대기 상태로 진입한다.

3.4 IVEF 서버 콘솔 프로그램 구현

IVEFResultTest 프로그램을 실행하기 위해서는 실행폴더에 있는 run.bat을 실행한다. IVEFResultTest 프로그램은 기존의 OpenIVEF SDK 프로그램과 새로 개발된 EtriIVEF library의 수행 결과를 비교하여 정상동작 유무를 검증하는 프로그램으로써, 실행 화면은 그림 9와 같다. 순차적으로 9개의 V-145 권고안에 따른 메시지에 대해 OpenIVEF SDK 라이브러리와 EtriIVEF 라이브러리의 결과 메시지를 비교하고 그에 따른 메시지 매칭 여부의 결과를 화면상에 디스플레이 한다. 라이브러리 검증이 끝난 후에는 AES 알고리즘의 테스트가 이루어진다. 메시지가 정상적으로 암호화 복호화 되는지 확인하고 이에 따른 결과를 화면상

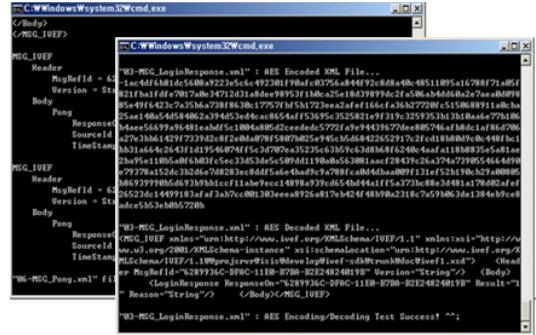


그림 9. IVEF 서버 콘솔 프로그램 실행 화면
Fig. 9. Screen of IVEF Server Console Program

에 표시한다.

IV. 구현 테스트 및 수행결과

OpenIVEF에서 제공하는 SAX 파서와 기존의 DOM 기반 EIVEF_Parser 및 EIVEF_PullParser의 성능 비교를 위해 PC 상에서 수행했을 때와 스마트폰에서 수행했을 때의 수행시간과 메모리 사용량을 비교하였다^[20]. 소스코드 폴더에 있는 IVEFPerf Test는 PC상에서 실행시간과 메모리 사용량을 측정하는 프로그램이며, Android 폰의 테스트 경우에는 Context 메뉴를 이용하여 테스트가 가능하다. 실증 테스트를 위한 안드로이드 폰으로는 LG전자의 옵티머스(Optimus) GK 폰을 이용하였다.

4.1 테스트 방법

테스트를 위해 V-145 규격에 따른 Track 데이터와 Vessel 데이터를 가지는 ObjectData가 200개, 400개, 600개, 800개, 1000개 포함된 ObjectData 메시지를 작성하여 이용하였다. 해당 메시지들에 대해 OpenIVEF 코드와 EIVEF_Parser, EIVEF_PullParser를 수행한 후, 파서가 수행되기 전후의 실행시간과 메모리 사용량을 비교 하였다. 유의할 점으로 PC의 경우 테스트 파일을 읽어 들일 때 빈번한 HDD I/O로의 접근이 발생하므로 테스트 프로그램 수행 후 많은 시간이 소요된다. 이에 비해 안드로이드 폰의 경우 모든 데이터가 메모리에 저장되므로 상대적으로 I/O 시간이 적게 소요된다. 테스트 결과부분의 성능비교 그림들은 각 해당 파서 수행시간 측정 코드를 보여준다.

4.2 테스트 결과

4.2.1 PC상에서의 성능측정 결과(ms)

PC 상에서의 수행 시간 측정 결과를 보면, EIVEF_PullParser가 수행시간이 가장 짧은 것을 알 수 있다. 일반적으로 SAX파서가 라인 인터프리트방식으로 일반적인 XML 파서 종류들 중에서 큰 문서를 핸들링 하고자 할 때나 빠른 처리를 요할 때 장점이 있으므로, SAX 파서가 수행시간이 더 짧을 것이라고 예상하였는데, 실제 수행결과는 Pull 파서가 더 짧게 나왔다. SAX의 경우 노드 이벤트를 모두 거치게 되고, Pull 파서의 경우 필요한 데이터가 추출되면 더 이상 이벤트를 거치지 않게 되어 수행시간이 짧게 걸린 것 같다.

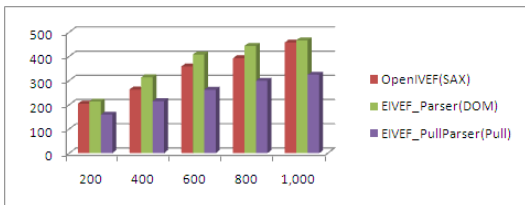


그림 10. PC상에서의 파서별 실행시간 측정 결과
Fig. 10. Performance Time on PC for Each Parser

4.2.2 PC상에서의 메모리 사용량 비교(kb)

메모리 사용량의 경우, 예상대로 DOM이 메모리 사용량이 가장 많았다. Pull 파서의 경우가 사용량이 가장 적었으며, SAX 파서가 의외로 메모리 사용량이 많았다. 내부적으로 이벤트 발생을 위해 필요한 구조체들과 빈번한 함수 호출로 인해 메모리 사용량이 많아진 것 같다. 또한 가비지 컬렉션이 제때에 이루어지지 않아 실제 사용하고 있지 않은 더미(Dummy) 메모리까지 같이 측정된 것 같다.

4.2.3 스마트폰 상에서의 성능측정 결과(ms)

안드로이드 폰 상에서의 테스트 결과도 전반적으로 Pull 파서의 성능이 좋게 나왔다. 특이한 점으로 PC

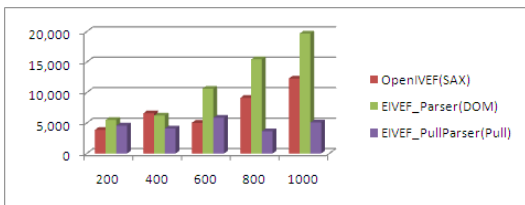


그림 11. PC상에서의 파서별 메모리 사용량 측정 결과
Fig. 11. Memory usage on PC for Each Parser

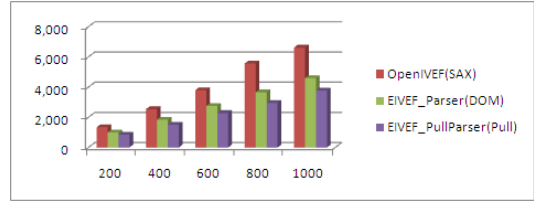


그림 12. 폰 상에서의 파서별 실행시간 측정 결과
Fig. 12. Performance Time on Phone for Each Parser

상에서의 테스트 결과와는 다르게 SAX 파서보다 DOM 파서의 성능이 더 좋게 나왔다. SAX 파서의 경우 빈번한 함수 호출이 발생하고, 안드로이드 플랫폼 상에서의 함수 콜 처리에 시간이 지연되어 이런 현상이 발생한 것 같다.

4.2.4 스마트폰상에서의 메모리 사용량 비교(kb)

안드로이드 폰 상에서 각각의 파서를 수행했을 때의 메모리 사용량을 보면, DOM 파서의 경우 객체가 1,000개 일 때 약 15.6MB까지 메모리 사용량이 증가되는 것을 알 수 있다. 이에 비해 XmlPullParser의 경우에는 1000개의 경우에도 메모리 사용량이 6M 정도인 것을 알 수 있다. 예상했던 것과는 달리 SAX의 경우 메모리 사용량이 계속 증가가 됐는데, 이는 상태를 관리하기 위해 사용했던 메모리가 적절히 해제되지 않아 이런 문제가 발생한 것 같다. 실제로는 가비지 메모리에 해당하므로 Pull 파서와 비슷한 메모리를 사용할 것으로 예상된다.

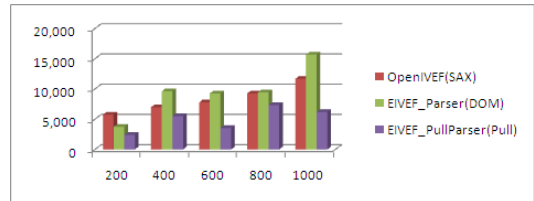


그림 13. 폰 상에서의 파서별 메모리 사용량 측정 결과
Fig. 13. Memory usage on Phone for Each Parser

V. 결 론

선박운항의 안전성 제고와 효율적인 해상교통 통제를 위해 선박기술 분야에 최신 전자 기술과 IT기술이 도입되고 있다. e-Navigation에서는 VTS 센터간의 데이터 교환을 위한 표준으로서 IVEF 서비스를 채택하고 있다. IVEF는 현재 개발이 진행 중인 프로젝트로서 아직 실제 구현과 성능의 검증이 완료되지 않았다. 본 논문에서는 DOM 파서와 SAX 파서의 중간 형

태를 가지는 XmlPullParser를 이용하여 기존 EIVEF_Parser와 동일한 인터페이스를 가지는 EIVEF_PullParser를 구현하였다. 그리고 EIVEF_PullParser와 기존 OpenIVEF파서, EIVEF_Parser와의 수행속도 및 메모리 사용량을 비교하였다. 테스트 결과 EIVEF_PullParser가 기존 OpenIVEF 파서나 EIVEF_Parser에 비해 수행속도가 빠르며, 메모리 사용량도 적게 소비된다는 결론을 얻었다. 향후 V-145 프로토콜의 구현에 있어 EIVEF_PullParser를 이용하는 것이 더 바람직 할 것으로 보인다.

본 논문에서는 안드로이드 플랫폼이 탑재된 휴대폰 상에 IVEF 클라이언트를 구현하였고, 데스크 탑 컴퓨터 상에 IVEF 서비스 서버를 구현하였다. 이를 통해 실제 선박과 VTS 센터 간의 상호 데이터 교환 상황을 에뮬레이션 할 수 있었으며, IVEF 서비스의 현업에서의 적용 가능성 여부를 예측할 수 있었다. 현재의 IVEF 서비스는 보안에 대한 절차나 정책이 표준으로 정의되어 있지 않아 이에 대한 연구가 필요하며, 향후 IVEF 프로토콜을 개선하여 레이더 정보와 CCTV 영상 정보도 효율적으로 제공할 수 있는 프로토콜의 설계가 필요할 것이다.

References

- [1] IALA Recommendation V-145 on the Inter-VTS Exchange Format(IVEF) Service, Jun. 2011.
- [2] WIKIPEDIA, E-Navigation, Retrieved Aug., 30, 2013, from <http://en.wikipedia.org/wiki/E-Navigation>
- [3] OpenIVEF Open Project(Release 1.0.5), Open IVEF(2010), Retrieved Oct., 30, 2013, from <http://www.openivef.org>.
- [4] International Association of Lighthouses and Aids-to-Navigation Authorities (IALA), "Interface control document for IVEF," Release 0.1.4, Retrieved Oct., 30, 2013, from http://code.google.com/p/ivef-sdk/source/browse/ivef-def/tags/IVEF_0_1_4/specs/ivef0_1.doc
- [5] B.-G. Lee, J.-W. Han, H.-S. Jo, N. Park, "A security architecture of the inter-VTS system for shore side collaboration of e-navigation," *J. Navigation and Port Research*, vol. 36, no. 1, pp. 1-7, 2012.
- [6] J. Lee, "Design and implementation of efficient mobile e-book viewer using mobile app framework," Gachon University Graduate School, 2012.
- [7] A. Developers, Introduction to Android, Retrieved Aug. 30, 2013, from <http://developer.android.com/guide>.
- [8] G. Seom and S. Seo, "Embodiment direction of Next e-Navigation," *J. Inst. Electron. Inf. Eng.*, vol. 34, no. 11, pp. 37-45, 2007.
- [9] B.-G. Lee, J.-W. Han, and H.-S. Jo, "Design of situation awareness and aids to navigation structure of VTS for maritime safety," *J. KICS*, vol. 35, no. 7, pp. 1073-1080, 2010.
- [10] K. Kim and N. Park, "Implementation of IVEF protocol between VTS systems using wireless communication on smart phone," *Korea Assoc. Inf. Education Research J.*, vol. 3, no. 3, pp. 93-98, Jeju, Korea, 2012.
- [11] N. Park, Y. Song, and K. Park, "Secure distributed data management architecture using AONT encryption in smart grid environment," *J. Korea Contents Assoc.*, vol. 10, no. 9, pp. 57-67, Sept. 2010.
- [12] S. Cho, J.-W. Jung, N. Park, B.-G. Lee, and D. Won, "Using the main authentication server and the IVEF in VTS system for improve of client authentication," *Korea Soc. Comput. Inf. Summer Conf. 2013*, vol. 20, no. 2, pp. 343-344, Busan, Korea, 2012.
- [13] N. Park, S. Cho, B.-D. Kim, B. Lee, and D. Won, "Security enhancement of user authentication scheme using IVEF in vessel traffic service system," *Lecture Notes in Electrical Eng.*, vol. 203, pp. 699-705, 2012.
- [14] K. Kim, B.-D. Kim, B. Lee, and N. Park, "Design and implementation of IVEF protocol using wireless communication on android mobile platform," *Commun. Comput. Inf. Sci.*, vol. 339, pp. 94-100, 2012.
- [15] N. Park, "The implementation of open embedded S/W platform for secure mobile RFID reader," *J. Korean Inst. Commun. Sci.*, vol. 35, no. 5, pp. 785-793, May 2010.
- [16] N. Park, "Implementation of personalized advertisement and information application

services using RFID virtual tag,” *J. Korea Soc. IT Services*, vol. 8, no. 4, pp. 151-163, Dec. 2009.

- [17] T. Kang and N. Park, “Design of J-VTS middleware based on IVEF Protocol,” *Lecture Notes Comput. Sci.*, vol. 7861, pp. 723-729, 2013.
- [18] N. Park, “User privacy preserving mobile RFID personal information security service system,” *J. Korean Inst. Inf. Technol.*, vol. 8, no. 10, pp. 87-96, Oct. 2010.
- [19] B. Lee and N. Park, “Security architecture of inter VTS exchange format protocol for secure u-Navigation,” *Lecture Notes Electrical Eng.*, vol. 181, pp. 229-236, 2012.
- [20] K. Kim, J. Kim, B.-G. Lee, and N. Park, “Performance enhancement of wireless IVEF protocol using XmlPullParser based on Android Mobile Platform,” *2013 KISS Fall Conf.*, pp. 350-352, Nov. 2013.
- [21] N. Park, “UHF/HF dual-band integrated mobile RFID/NFC linkage method for mobile device-based business application,” *J. Korean Inst. Commun. Sci.*, vol. 38, no. 10, pp. 841-851, Oct. 2013.
- [22] N. Park and Y. Song, “Secure distributed data management architecture using AONT encryption in smart grid environment,” *J. Korean Inst. Commun. Sci.*, vol. 35, no. 10, pp. 1458-1470, Oct. 2010.
- [23] N. Park, “Analysis of privacy weakness and protective countermeasures in smart grid environment,” *J. Korean Inst. Inf. Technol.*, vol. 8, no. 9, pp. 189-197, Sept. 2010.

박 남 제 (Namje Park)



2000년 8월 : 동국대학교 정보산업학과 졸업

2003년 8월 : 성균관대학교 정보보호학과 석사

2008년 3월 : 성균관대학교 컴퓨터공학과 박사

2003년 4월~2008년 12월 : 한국전자통신연구원 선임연구원

2009년 1월~2009년 12월 : University of California at Los Angeles (UCLA) Post-Doc.

2009년 3월~2009년 12월 : University of California at Los Angeles (UCLA) WINMEC 연구센터 Staff Researcher

2010년 1월~2010년 9월 : Arizona State University (ASU) Research Scientist

2010년 9월~현재 : 제주대학교 교육대학 컴퓨터교육전공 교수

<관심분야> 암호이론, 컴퓨터교육, 융합기술보안, 모바일컴퓨팅, RFID/USN, 스마트그리드 보안 등