

WAVE와 CAN 연동을 위한 OSEK OS기반 게이트웨이

김주영*, 서현수*, 이상선^o

OSEK OS Based Gateway for Interconnecting WAVE and CAN

Ju-young Kim*, Hyun-soo Seo*, Sang-sun Lee^o

요약

최근 차량간 통신 프로토콜 표준인 WAVE를 이용한 다양한 서비스가 이루어지고 있으며, 안정적인 시스템 운영을 위해서 기존의 차량 내부 네트워크와의 원활한 연계를 위한 게이트웨이가 필요하다. 따라서, 본 논문에서는 차량 외부 통신의 대표적인 프로토콜인 WAVE와 차량 내부 네트워크의 대표적인 프로토콜인 CAN을 연동하기 위한 게이트웨이를 제안한다. 제안하는 게이트웨이는 OSEK 운영체제 기반으로 설계되었으며, 통신 계층, 메시지 변환 계층, 메시지 관리 계층으로 구성된다. 통신 계층은 CAN을 통해 차량 정보를 수신하거나, WAVE 모듈로부터 주행안전정보 요청 메시지를 수신, 주행안전정보를 송신한다. 그리고 메시지 관리 계층에서는 수신 메시지 저장, 메시지의 유효성에 대한 오류검사 기능을 수행한다. 게이트웨이에 대한 성능분석을 위해 두 가지의 테스트 시나리오를 구성하였다. 첫 번째는 차량간 통신을 이용한 안전서비스에서의 메시지 구조를 정의한 BSM의 주기적인 전송이 이루어질 경우에서의 성능 테스트이다. 두 번째는 노변장치가 주행안전정보를 요청하는 경우에 게이트웨이가 차량내부정보를 응답하는 경우이다. 테스트 결과로써, 게이트웨이 자체 처리시간은 각각 0.32ms, 0.46ms으로 안정적으로 동작함을 확인하였다.

Key Words : WAVE, CAN, Gateway, OSEK OS, V2V communications

ABSTRACT

Recently, various services are provided by using WAVE protocol for communication among neighboring vehicles. And in order to operate stable system, the gateway for interconnecting in and out vehicle networks is required. In this paper, we propose gateway interconnecting WAVE and CAN protocol. The proposed gateway based on OSEK OS consists of a communication layer, a message translator layer and a message management layer. In the communication layer, WAVE communication part and CAN communication part are designed to communicate with WAVE and CAN. And in the message management layer, message management layer functions to store the received messages and check errors with the message. Based on these functions, experiment was conducted to analyze performance of the gateway with two scenarios such as transmitting periodically BSM as a message structure for safety services in vehicle-to-vehicle communications and responding to road side equipments requiring in-vehicle information. As a result of test, we verify our gateway performance by analyzing measured time in test scenarios.

※ 본 연구는 국토해양부 지원을 받는 건설기술혁신사업의 지원으로 수행되었습니다.(스마트하이웨이사업(07 기술혁신 A01))

♦ First Author : 한양대학교 전자컴퓨터통신공학과 유비쿼터스통신 연구실, kjuyoung@nate.com, 학생회원

° Corresponding Author : 한양대학교 융합전자공학부 유비쿼터스통신 연구실, ssnlee@hanyang.ac.kr, 종신회원

* 한양대학교 전자컴퓨터통신공학과 유비쿼터스통신 연구실, digitalshs@hanyang.ac.kr, 학생회원

논문번호 : KICS2014-02-038, 접수일자 : 2014년 2월 4일, 심사일자 : 2014년 2월 13일, 최종논문접수일자 : 2014년 2월 24일

I. 서 론

최근 차량간 통신 프로토콜인 WAVE를 기반으로 다양한 서비스들이 이루어지고 있다. 대표적인 차량간 통신 프로토콜인 WAVE(Wireless Access in Vehicular Environments)는 WLAN(Wireless Local Area Network)을 기반으로 간결한 연결성과 우수한 이동성의 특징을 가지며, 2010년 IEEE에서 제정된 표준이다¹⁻³⁾. 차량간 통신을 이용한 서비스는 차량 내부 정보에 추가적으로 차량간 통신을 통해 수집한 주변 정보까지 활용함으로써, 날씨 및 장애물에 대해 강인한 성능을 가질 수 있다. 특히, 유럽의 Drive C2X, 미국의 Connected Vehicle, VSC-A(Vehicle Safety Communication-Applications) 프로젝트에서 전방추돌 방지, 안전 추월 보조, 차선변경 등 다양한 서비스에 대해 대규모 테스트를 진행하고 있으며, 관련 법규 제정을 위한 연구가 수행되고 있다⁴⁻⁶⁾.

또한, 차량에는 다양한 센서 등이 차내망을 통해 연결되어 운전자의 안전과 편의를 위한 다양한 서비스가 제시되고 있다⁷⁾. 특히, ADAS(Advanced Driver Assistance System) 등 능동 안전 시스템에서는 초음파, 비전 등 사물을 인지하기 위한 센서들이 차량에 부착되어 미리 사고를 방지하여 운전자의 안전을 도모한다. 많은 센서들과 장치들이 차량에 부착됨에 따라, 다양한 종류의 차내망 기술이 발전되어 왔다. 대표적인 차내망 프로토콜로는 LIN(Local Interconnect Network), CAN(Controller Area Network), MOST(Media Oriented Systems Transport) 등이 ECU(Electronic Control Unit)의 특성에 따라 구성되어 사용되고 있다. 최근 대용량 멀티미디어 장치와 카메라의 차량 장착으로 인해 동기식 이더넷이 활용되고 있다⁸⁾. 그리고 AUTOSAR(AUTomotive Open System ARchitecture)를 비롯한 차내내 플랫폼이 구성되고 적용되고 있으며, OSEK OS(Open System Electronics in Vehicles Operating System)등과 같은 차량용 운영체제가 적용되고 있는 상황이다⁹⁾.

따라서 차량환경에서 다양한 네트워크를 연동하기 위한 연구들이 진행되어 왔다. CAN, LIN, FlexRay 프로토콜을 사용하는 장치와 MOST 프로토콜을 사용하는 장치 간의 원활한 동작을 위해 애플리케이션 레벨의 게이트웨이를 구현하였다¹⁰⁾. 또한 최근 발전하고 있는 휴대용 스마트 기기와 차량 내부 네트워크 프로토콜인 CAN과 연동하여 차량 정보 요청, 전조등 제어 서비스를 제공하기 위해 게이트웨이를 구현하였다¹¹⁾. 그러나 기존에 수행된 연구에서는 차내망 또는

스마트기기와 차량간 인터페이스에 대한 연구이며, 차량간 통신을 통해 수집되는 정보를 차내내부 네트워크와 연동하기 위한 연구는 현재까지 수행되지 않았다. 따라서 본 논문에서는 차량간 통신 프로토콜인 WAVE와 차내망 네트워크인 CAN간 인터페이스를 위한 게이트웨이를 제안하였다. 제안하는 게이트웨이는 OSEK 운영체제의 한 종류인 Trampoline을 기반으로 구현되었으며, 통신 계층, 메시지 변환 계층, 메시지 관리 계층으로 구성된다. 통신 계층은 CAN 통신부를 통해 차량 정보를 수신하거나, WAVE 통신부를 통해 WAVE 모듈로부터 주행안전정보 요청 메시지를 수신, 주행안전정보를 송신한다. 그리고 메시지 관리 계층에서는 수신 메시지 저장, 메시지의 유효성에 대한 오류검사 기능을 수행한다. 성능 측정을 위해서는 두 가지 상황을 구성하였다. 주변 차량에 주행안전정보를 담은 BSM(Basic Safety Message)을 주기적으로 전송하는 상황과 노변장치로부터의 주행안전정보 요청에 대해 응답을 전송하는 상황을 구성하여 시뮬레이션하였다. 성능 분석을 위해서는 메시지 처리 시간, CAN 시뮬레이터와 WAVE-CAN 게이트웨이 그리고 WAVE 모듈간 전송 시간을 측정하여 분석 및 평가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 통신 인터페이스 설계를 위해 관련 기술인 WAVE와 CAN를 소개하고, 차량 운영체제 환경을 이해하기 위한 OSEK 운영체제에 대해 소개한다. 3장에서는 게이트웨이 시스템 구조 및 주요 기능, 흐름에 대해 제시한다. 4장에서는 게이트웨이의 동작 검증 및 성능평가를 나타내며, 마지막으로 5장에서는 결과를 정리하고 결론을 도출한다.

II. 본 론

제안하고자 하는 게이트웨이에 적용된 요소 기술로는 차량 외부 통신 프로토콜인 WAVE, 차량 내부 네트워크 프로토콜인 CAN 그리고 차량용 운영체제인 OSEK 운영체제가 존재한다. 본 장에서는 WAVE와 CAN, OSEK 운영체제에 대해 간략하게 설명하고자 한다.

WAVE는 WLAN 802.11을 기반으로 빠른 이동성, 연결성을 요구하는 차량환경에서 데이터를 송수신할 수 있는 차량간 통신 표준 프로토콜으로써 IEEE에서 제정되었다¹¹⁻¹³⁾. WAVE 아키텍처를 그림 1에 나타내었으며, 물리 계층과 데이터 링크 계층을 정의하는 IEEE Std 802.11p, 채널 스위칭 담당과 상위 MAC 계

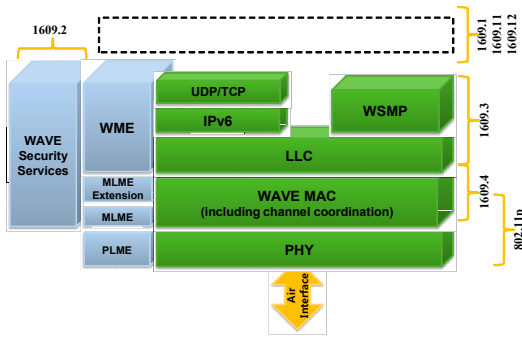


그림 1. WAVE 아키텍처
Fig. 1. WAVE architecture

층을 포함하는 IEEE Std 1609.4, 상위 애플리케이션 계층에 연결해주는 네트워킹 서비스를 규정하고 IPv6 기반 네트워크 계층을 포함하는 IEEE Std 1609.3이 존재한다. 그리고 보안을 정의하는 IEEE Std 1609.2 및 원격 관리와 관련된 IEEE Std 1609.1이 존재한다.

차량 내부 환경에서는 다양한 전자장치들이 복잡하게 연결됨에 따라 배선의 수를 감소시키고 효율적인 통신을 위해 1983년 보쉬에서 ECU간 또는 전자장치간 다중 접속 통신 프로토콜인 CAN을 개발하였으며, 아키텍처는 그림 2와 같다^[12]. CAN 네트워크의 구성은 CAN 네트워크 내에서 신호가 전송되는 방식을 결정하고 신호레벨과 비트 표현방법, 전송매체 등을 정의하는 물리 계층과 수신 메시지 필터링, 데이터 전송 및 요청하는 서비스를 제공하는 데이터 링크 계층의 LLC(Logical Link Control) 하부계층이 존재한다. 그리고 메시지 프레임 제어, 에러 검출 및 에러신호 전송, CAN 데이터 중재 및 응답을 담당하는 CAN의 핵심계층인 MAC(Medium Access Control) 하부계층과

Application Layer	
Data Link LLC sublayer	Acceptance filtering Overload notification Recovery management
MAC sublayer	Data encapsulation/decapsulation Frame coding (stuffing/destuffing) Medium access management Error detection Error signaling Acknowledgement Serialization/Deserialization
Physical	Bit encoding/decoding Bit timing Synchronization
Driver/Receiver characteristics	

그림 2. CAN 아키텍처
Fig. 2. CAN architecture

애플리케이션 계층으로 구성된다. OSEK은 1993년 BMW, 보쉬, 다임러 등 독일의 자동차 업체들이 컨소시엄을 구성하여 진행한 프로젝트이다. 차량용 공개 시스템 및 인터페이스 표준을 뜻하며, 운영체제(OS), 통신 프로토콜 스택(COM), 네트워크 관리 프로토콜(NM) 등이 포함되어 있다. 그 중에서 OSEK 운영체제는 차량환경에 적합한 형태로 개발된 실시간 운영체제이다^[13,14]. 현재 OSEK 운영체제는 다양한 전자장치와 차내망의 효율적인 운영을 위해 사용되고 있으며, AUTOSAR 운영체제의 기반 기술로도 사용되고 있다.

III. WAVE-CAN 게이트웨이 시스템

본 장에서는 제안한 게이트웨이의 구조와 기능, 구성 및 흐름에 대해 정의한다.

3.1. WAVE-CAN 게이트웨이 구조

게이트웨이는 크게 통신 계층(Communication layer), 메시지 변환 계층(Message Translator layer), 메시지 관리 계층(Message Management layer)으로 구성되며, 그림 3에 게이트웨이 구조를 도식화하였다. 게이트웨이 내외부 통신을 위한 통신 계층은 CAN 통신부, WAVE 통신부로 구성된다. 본 논문에서는 차량 내부정보를 생성하는 CAN 시뮬레이터를 구성하였으며, CAN 통신부와 연계되어 차량정보를 주고받을 수 있게 하였다.

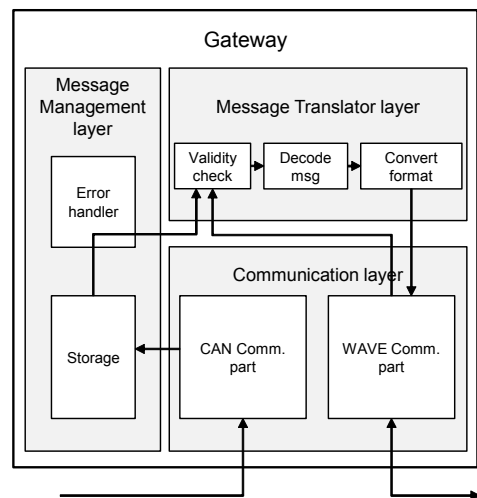


그림 3. WAVE-CAN 게이트웨이 구조
Fig. 3. WAVE-CAN gateway architecture

그리고 WAVE 통신부는 WAVE 통신을 통해 다른 노드와 BSM 안전 정보를 송수신할 수 있게 하였다. BSM은 차량간 통신을 이용한 안전 서비스 관련 정보 전송을 위해 미자동차공학회에서 정의한 SAE(Society of Automotive Engineers) J2735 표준을 나타내고, BSM의 part1의 구조는 그림 4와 같다¹⁵⁾. 본 논문에서는 일반적인 차량 상태를 part1 부분에 삽입하여 전송한다. 엔진, 새시 등의 정보는 운전자의 안전과 밀접하기 때문에 이런 장치들을 제어들은 불가능하도록 하였으며, 읽는 기능만 가능하도록 설계하였다.

메시지 변환 계층은 WAVE와 CAN 통신 사이에 송수신이 가능하도록 각각의 메시지 형태에 맞게 변환해주는 역할을 한다. 세부적인 기능으로써, 저장, 유효성 검사를 통해 수신 메시지가 정상적인 메시지로 확인되면 식별자, 데이터 길이, 데이터와 같은 필요한 정보를 추출하여 다른 프로토콜의 메시지 형태로의 변환을 준비한다. 그리고 추출된 정보를 기반으로 BSM 데이터 요소 형태와 CAN 메시지 형태 간 알맞은 메시지 형태 변환을 수행하며, 이를 그림 5에 도식화하였다. 그림으로부터, 수신 메시지의 헤더와 체크섬을 통해 유효성을 검사하고 메시지 순서를 나타내는 번호, CAN 메시지 정보인 식별자, 데이터 길이, 데이터를 추출한다. 그리고 외부에서 수신된 요청 어

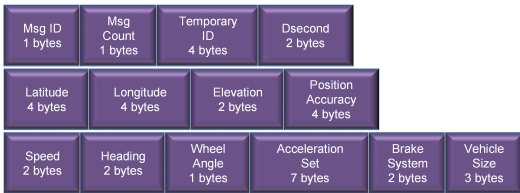


그림 4. BSM part 1 구조도
Fig. 4. BSM part 1 structure

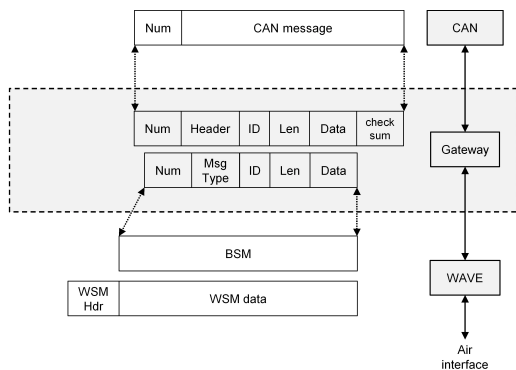


그림 5. 메시지 변환 과정
Fig. 5. Message conversion process

부를 식별할 수 있는 Msg Type 필드를 추가한 후 BSM을 전송한다.

메시지 관리 계층에서는 통신 계층을 통해 메시지를 수신할 때 메시지의 입력과 출력처리에서 속도차이가 발생할 수 있기 때문에 처리 속도차이를 완화시킬 수 있는 저장 기능을 제공하고, 메시지들이 순서대로 처리될 수 있도록 차례대로 메시지를 읽어와 처리하는 기능을 제공한다. 그리고 메시지 송수신 시 또는 메시지 형태 변환 이전에 발생 가능한 오류를 처리하는 기능을 제공한다. 오류 처리 과정에서 식별자에 대해 필터링 하여 유효하지 않은 메시지는 폐기하고, 유효한 메시지는 체크섬을 통해 오류가 없는지 확인한다. 이후에 정상적인 메시지는 메시지 변환 과정을 거쳐 WAVE 모듈로 전달된다. 마지막으로 파일에 주요 함수의 실행시간을 출력하여 처리 시간, 전송 시간을 측정하고 성능 분석 및 평가할 수 있게 하였다.

3.2 WAVE-CAN 게이트웨이 프로세스

3.2.1 게이트웨이 프로세스 구조

게이트웨이는 두 개의 프로세스 구조로 이루어진다. 첫 번째는 CAN 시뮬레이터에 연결되어 차량 정보가 포함된 CAN 메시지를 수신하는 프로세스이며, 두 번째는 WAVE 통신을 하는 노변장치를 연결되어 주행안전정보 요청과 응답을 송수신하는 프로세스이다. 이러한 두 가지 프로세스의 흐름도를 그림 6에 나타내었다. 첫 번째 프로세스는 차량내부 ECU가 게이트웨이를 통해 외부에 정보를 전송할 경우에 대한 시나리오를 수행하는 것이며, CAN 시뮬레이터로부터 수신된 CAN 정보에 대한 유효성 확인, 메시지 형태 변환 과정을 거쳐 WAVE 모듈에 전달한다. 그리고 두 번째 프로세스는 노변장치가 WAVE를 통해 차량 정보에 대한 요청이 있을 경우를 처리하기 위한 과정이며, 노변장치로부터의 주행안전정보 요청을 WAVE 모듈을 통해 게이트웨이가 수신하면, 요청에 대한 응답에 필요한 정보를 구성한 후 WAVE 모듈을 통해 주행안전정보를 전송한다.

CAN 시뮬레이터에서 차량의 정보 데이터를 생성할 수 있도록 구성하였으며, 게이트웨이와의 연결은 가상 시리얼 통신을 통해 이루어진다. CAN 메시지 프레임은 CAN-A 표준문서를 바탕으로 11비트의 식별자를 가지며, CAN 시뮬레이터로부터 메시지가 생성되면 식별자 값에 의해 목적지가 정해진다. 그리고 시뮬레이터 내의 체크섬 함수를 통해 데이터 프레임이 제대로 전달될 수 있도록 설계하였다. WAVE 모

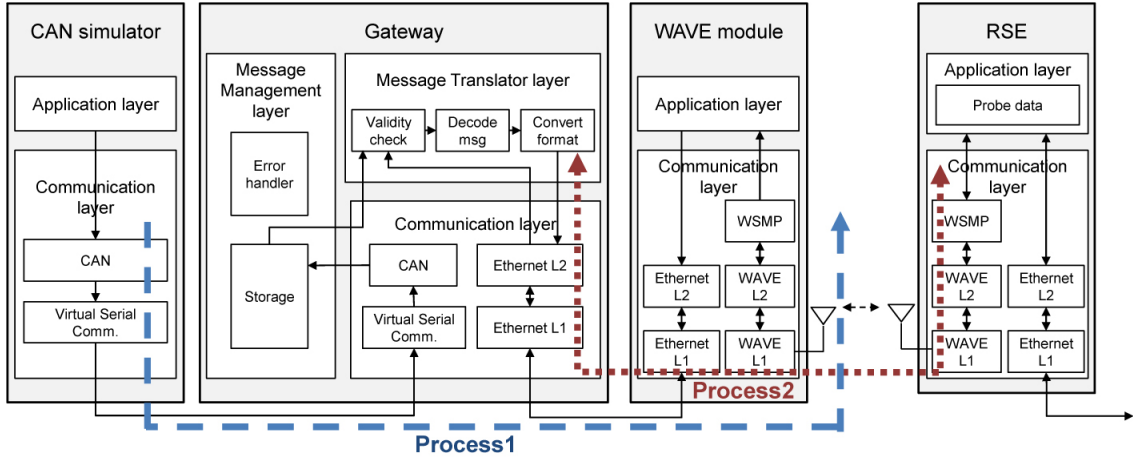


그림 6. 전체 시스템 프로세스 구조
Fig. 6. Gateway system process architecture

들은 IEEE std 1609.3, IEEE std 1609.4, 802.11p 표준이 구현되어 있으며, 5GHz 대역의 주파수를 통해 다른 노드와 WSM(WAVE Short Message)을 송수신한다. 게이트웨이와 WAVE 모듈은 UDP(User Datagram Protocol)를 통해 연결되어 있으며, 이를 통해 차량 내외부 데이터를 주고받는다. WAVE 모듈은 게이트웨이로부터 수신한 주행안전정보를 BSM 형태로 외부의 다른 노드로 송신하거나, 다른 WAVE 모듈로부터 수신한 필요정보 요청 메시지를 게이트웨이로 송신하는 전달자 역할을 수행한다.

3.2.2 게이트웨이 프로세스 흐름

설계한 게이트웨이 프로세스 흐름은 두 가지로 나뉜다. 첫 번째는 그림 7과 같이, CAN 시뮬레이터로부터 자기 차량의 속도, 가속도, 진행방향, 브레이크 상태, 위치 등 안전 서비스 관련 CAN 정보를 주기적으로 수신하여 관리하며, WAVE 통신을 위한 BSM을 구성하여 100ms 주기로 주변 노드에 전송하는 과정

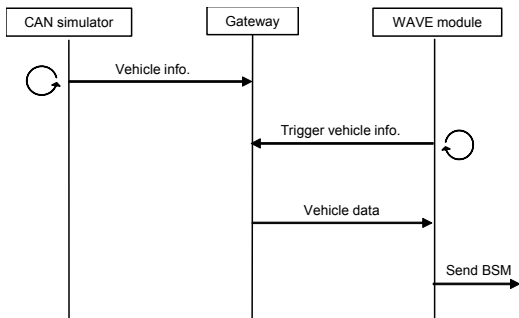


그림 7. 주기적인 BSM 전송
Fig. 7. Sending periodically BSM

이다.

두 번째 흐름은 그림 8과 같이 노변장치로부터 WAVE 모듈을 통해 주행안전정보 요청 메시지를 수신하게 되면, 와이퍼 상태, 레인 센서값, 안개등 상태 등의 주행안전정보를 응답하는 과정이다. 게이트웨이는 CAN 시뮬레이터로부터 차량 정보를 주기적으로 수신하여 저장하고 있으며, 주행안전정보 요청을 수신하면 저장 공간에서 요청된 정보에 대해 검색한다. 그리고 검색이 되면, 최신 정보를 추출하여 BSM을 생성한 후 응답한다.

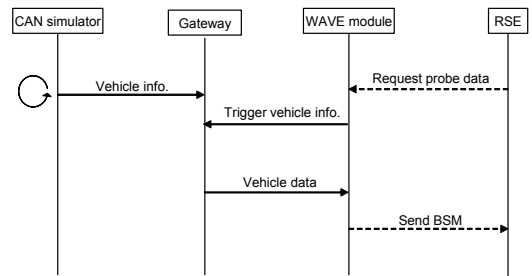


그림 8. 노변장치로부터 주변 차량의 정보 수집을 위한 요청
Fig. 8. Request to collect information of vehicles from RSE

IV. 실험 환경 및 성능 평가

본 게이트웨이는 OSEK 운영체제를 기반으로 구현되었으며, 성능평가를 위한 실험환경은 그림 9와 같다. 개인용 컴퓨터에 게이트웨이와 CAN 시뮬레이터가 탑재되어 있고, WAVE 모듈을 게이트웨이의 통신 계층에 연결하였다. 그리고 실험을 위해 6Mbps의 속

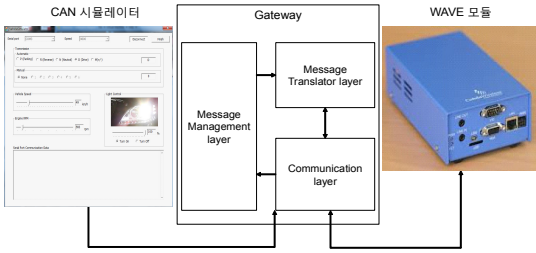


그림 9. 시스템 실험 환경 구성도
Fig. 9. System experiment configuration

도, 33dBm의 전력을 가지며, SCH 172번에서 데이터가 브로드캐스팅으로 전송된다. 게이트웨이는 OSEK OIL을 이용해 개발하였고 그림 10과 같이 OSEK OIL로 작성한 게이트웨이 내부 구조를 간략히 나타내었다. 각각의 태스크는 송신, 수신, 메시지 구조 변환 등의 처리를 담당하며, 총 6개로 구성하여 소프트웨어 구조를 단순화 시켰다. 게이트웨이 성능평가를 위해 다음의 두 가지 실험을 구성하였다. 첫 번째 실험은 차량 내부정보를 BSM에 삽입하여 주기적으로 전송하는 상황이다. 두 번째 실험은 노변장치로부터 주행 안전정보에 대한 요청 메시지를 수신하였을 때 게이트웨이에서 요청에 대한 응답을 하는 상황이다.

그림 11은 BSM을 주기적으로 전송하는 경우의 전송 시간과 처리 시간에 대한 관계를 나타낸다. 여기서 $T_{Gateway-process}$ 는 WAVE 모듈로부터 BSM에 삽입할 차량 내부정보 요청을 수신한 후 요청에 대한 정보를 송

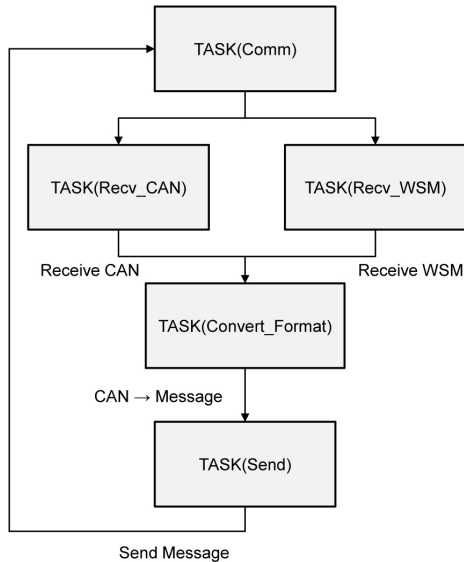


그림 10. OSEK OIL로 작성한 게이트웨이 구조
Fig. 10. Gateway structure written by OSEK OIL

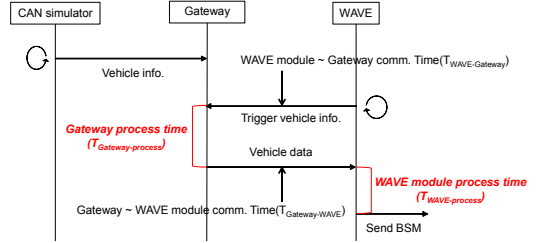


그림 11. BSM의 주기적인 전송 시 처리 시간 및 전송 시간의 관계
Fig. 11. Relation of process time and transmission time when sending periodically BSM

신하기 전까지의 처리 시간이고, $T_{WAVE-Gateway}$, $T_{Gateway-WAVE}$ 는 게이트웨이와 WAVE 모듈 간 전송 시간이다. 그리고 $T_{WAVE-process}$ 는 게이트웨이로부터 BSM에 삽입할 정보를 수신하고, BSM에 삽입한 후 주변 차량에 전송하기 전까지의 처리 시간이다.

첫 번째 실험결과를 표 1, 2와 그림 12에 정리하였다. 테스트 횟수는 100번이고 표 1은 실험결과의 평균값, 표 2는 실험결과 최악의 값을 나타낸 것이다. 표 1로부터, 주기적인 BSM을 전송하는 경우의 평균 $T_{Gateway-process}$ 는 0.32ms로 짧은 처리시간을 보이지만, 평균 $T_{WAVE-process}$ 는 5.32ms로 $T_{Gateway-process}$ 에 비해 상대적으로 긴 처리속도를 나타낸다. 이렇게 $T_{Gateway-process}$ 가 $T_{WAVE-process}$ 보다 짧은 이유는 게이트웨이를 실시간 운영체제 환경에서 구성하고, 소프트웨어 계층 구조를 간략화 하였으며, 계층별 기능을 단순화시켰기 때문이다. 또한 표 2에서 주기적인 BSM을 전송하는 경우 최악의 $T_{Gateway-process}$ 는 1.42ms로 평균 $T_{Gateway-process}$ 보다 길어진 것을 알 수 있다. 그 이유는 WAVE 모듈로 보낼 메시지를 생성하고 전송하는 과

표 1. 주기적인 BSM을 전송할 경우 평균 처리 시간 및 평균 전송 시간

Table 1. Average process time and average transmission time when sending periodically BSM

$T_{Gateway-process}$	$T_{WAVE-Gateway}$ + $T_{Gateway-WAVE}$	$T_{WAVE-process}$
0.32ms	3.94ms	5.32ms

표 2. 주기적인 BSM을 전송할 경우 최악의 처리 시간 및 최악의 전송 시간

Table 2. Worst process time and worst transmission time when sending periodically BSM

$T_{Gateway-process}$	$T_{WAVE-Gateway}$ + $T_{Gateway-WAVE}$	$T_{WAVE-process}$
1.42ms	5.03ms	6.66ms

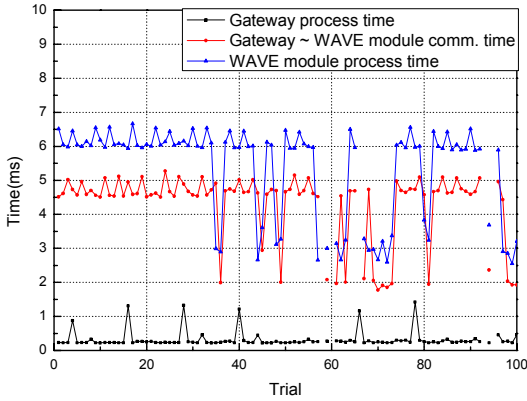


그림 12. BSM을 주기적으로 전송할 경우
Fig. 12. Case of sending periodically BSM

정에서 처리 시간이 증가하였기 때문이다. 그리고 최악의 $T_{WAVE-process}$ 는 6.66ms로 평균 $T_{WAVE-process}$ 보다 길어진 것을 알 수 있는데 BSM 생성, WSM 인코딩 과정에서 처리 시간이 증가하였기 때문이다. 그림 12의 가로축은 메시지 개수, 세로축은 시간을 나타낸다. 그래프에서 $T_{Gateway-process}$ 가 증가하는 지점이 발생하는 이유는 CAN 시뮬레이터가 간헐적으로 짧은 시간동안 많은 정보를 생성하고, 저장 장소에 저장한 많은 정보 중 가장 최신 정보를 검색, 처리하는 시간이 증가하였기 때문이다. 그리고 $T_{WAVE-process}$ 가 감소하는 지점이 발생하는 이유는 게이트웨이로부터 최소한의 정보를 수신하고, 이 정보를 기반으로 빠른 BSM 생성, WSM 인코딩이 이루어졌기 때문이다.

그림 13은 두 번째 실험인 노변장치의 요청에 대한 응답을 전송하는 경우의 전송 시간과 처리 시간에 대한 관계를 나타낸다. 여기서 $T_{WAVE-process}$ 는 WAVE 모듈이 노변장치로부터 주행안전정보 요청을 수신하고 그 요청을 WAVE-CAN 게이트웨이로 전달하기 전까

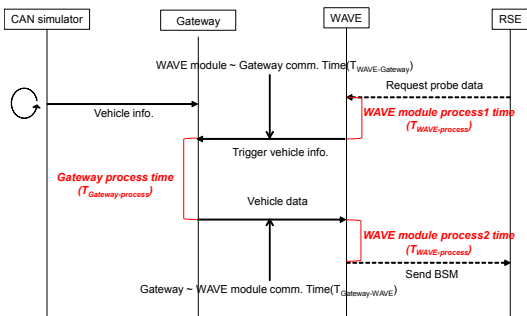


그림 13. 노변장치의 요청에 대한 응답 전송 시 처리 시간 및 전송 시간의 관계
Fig. 13. Relation of process time and transmission time when sending response to RSE

지의 처리 시간이고, $T_{WAVE-Gateway}$, $T_{Gateway-WAVE}$ 는 WAVE 모듈과 WAVE-CAN 게이트웨이 간 전송 시간이다. 그리고 $T_{Gateway-process}$ 는 WAVE-CAN 게이트웨이가 WAVE 모듈로부터 요청을 수신하고 요청된 주행안전정보를 WAVE 모듈로 송신하기 전까지의 처리 시간이다. 마지막으로 $T_{WAVE-process2}$ 는 WAVE 모듈이 게이트웨이로부터 주행안전정보를 수신하고 BSM에 삽입해 노변장치에 응답하기 전까지의 처리 시간이다.

두 번째 실험결과를 표 3, 4와 그림 13에 나타내었다. 표 3은 실험결과의 평균값, 표 4는 실험결과 최악의 값을 정리하였다. 표 3로부터, 첫 번째 실험인 주기적인 BSM을 전송하는 경우와 비교해서 $T_{Gateway-process}$ 와 $T_{WAVE-Gateway}$, $T_{Gateway-WAVE}$ 의 평균값은 유사함을 알 수 있으며, $T_{WAVE-process1} + T_{WAVE-process2}$ 는 증가한 것을 알 수 있다. 전체적인 $T_{WAVE-process1} + T_{WAVE-process2}$ 가 증가한 이유는 주기적인 BSM을 전송하는 상황을 기반으로 노변장치로부터 수신한 주행안전정보 요청에 대하여 응답하는 추가적인 처리 시간이 발생했기 때문이다. 그리고 표 4에서 노변장치의 요청에 응답을 전송하는 경우 최악의 $T_{Gateway-process}$ 는 평균 $T_{Gateway-process}$ 보다 길어진 것을 볼 수 있다. 주기적인 BSM을 전송할 경우와 마찬가지로 WAVE 모듈로 보낼 메시지를 생성하고 전송하는 과정에서 처리 시간이 증가하였기 때문이다. 또한 최악의 $T_{WAVE-process1}$ 는 평균 $T_{WAVE-process1}$ 과 비교하여 큰 차이가 없음을 알 수 있지만 최악의 $T_{WAVE-process2}$ 는 평균 $T_{WAVE-process2}$ 보다 증가한 것을 알 수 있다. 그 이유는 정보 수신, BSM 생성, WSM 인코딩 처리 시간이 증가하였기 때문이다. 그림 14 그래프의 $T_{Gateway-process}$ 가 증가하는

표 3. 노변장치의 요청에 응답을 전송하는 경우 평균 처리 시간 및 평균 전송 시간

Table 3. Average process time and average transmission time when sending response to RSE

$T_{Gateway-process}$	$T_{WAVE-process1}$	$T_{WAVE-process2}$	$T_{WAVE-Gateway} + T_{Gateway-WAVE}$
0.46ms	1.66ms	5.49ms	3.06ms

표 4. 노변장치의 요청에 응답을 전송하는 경우 최악의 처리 시간 및 최악의 전송 시간

Table 4. Worst process time and worst transmission time when sending response to RSE

$T_{Gateway-process}$	$T_{WAVE-process1}$	$T_{WAVE-process2}$	$T_{WAVE-Gateway} + T_{Gateway-WAVE}$
2.05ms	2.19ms	7.24ms	5.18ms

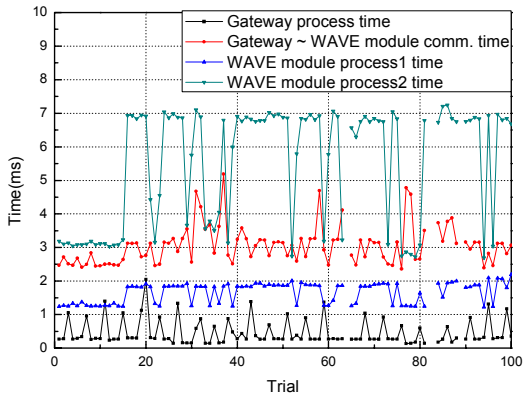


그림 14. 노변장치의 요청에 대한 응답을 전송하는 경우
Fig. 14. Case of sending response to 노변장치

지점이 자주 발생하는 이유는 CAN 시뮬레이터로부터 간헐적으로 수신하는 정보와 노변장치로부터 주기적으로 수신하는 주행안전정보 요청을 처리하기 때문이다. 또한 $T_{WAVE-process1}$ 와 $T_{WAVE-process2}$ 가 증가하는 지점이 자주 발생하는 이유는 노변장치로부터 주기적인 요청 수신, 게이트웨이로 요청 전달을 수행하고, 요청에 대한 응답으로 주행안전정보를 전송하는 처리 과정을 수행하기 때문이다. 두 가지 테스트 과정을 통해 게이트웨이 자체 처리속도인 $T_{Gateway-process}$ 가 각각 0.32ms, 0.46ms의 빠른 처리속도를 보임에 따라 차내 외부망을 연동하기 위한 게이트웨이로써 동작할 수 있을 것으로 판단된다.

V. 결론

본 논문에서는 차량간통신인 WAVE와 차내망인 CAN간의 유기적인 인터페이스 환경을 구성하기 위하여 OSEK 운영체제 기반의 게이트웨이를 설계 및 성능평가를 수행하였다. 제안하는 통신 계층, 메시지 변환 계층, 메시지 관리 계층으로 구성된다. 통신 계층은 CAN 통신부를 통해 차량 정보를 수신하거나, WAVE 통신부를 통해 WAVE 모듈로부터 주행안전정보 요청 메시지를 수신, 주행안전정보를 송신한다. 제안한 게이트웨이 성능 평가를 위해 CAN 시뮬레이터를 구성하였으며, 차량 내부정보를 BSM에 삽입하여 주기적으로 전송하는 상황과 노변장치로부터 주행안전정보에 대한 요청 메시지를 수신하였을 때 게이트웨이에서 요청에 대한 응답을 하는 상황을 가정하여 각 프로세스의 동작 시간을 측정하였다. 실험결과로써, 게이트웨이 자체 처리속도인 $T_{Gateway-process}$ 가 각각 0.32ms, 0.46ms를 나타냄에 따라 CAN 메시지 수

신, 게이트웨이와 WAVE 모듈 간 메시지 전달이 안정적으로 신속히 수행됨을 확인할 수 있었다. 향후 과제로는 다수의 WAVE 모듈을 이용한 성능 테스트가 이루어져야 하고, 정해진 서비스 시나리오뿐만 아니라 CAN 송수신에 대한 성능 테스트가 필요할 것으로 사료된다.

References

- [1] IEEE Std 802.11p, "IEEE standard for information technology-telecommunications and information exchange between systems- local and metropolitan area networks Specific requirements, part 11: Wireless LAN medium access control(MAC) and physical layer (PHY) specifications, amendment 6: Wireless access in vehicular environments," IEEE, Jul. 2010.
- [2] IEEE Std 1609.4, "IEEE standard for wireless access in vehicular environments(WAVE)-multi-channel operation," IEEE, Feb. 2011.
- [3] IEEE Std 1609.3, "IEEE standard for wireless access in vehicular environments(WAVE)-networking services," IEEE, Dec. 2010.
- [4] H. S. Seo, J. S. Jung, and S. S. Lee, "Scenario and network performance evaluation for a do not pass warning service based on vehicle-to-vehicle communications," *J. KICS*, vol. 38C, no. 3, pp. 227-232, Mar. 2013.
- [5] Connected Vehicle Project, <http://icsw.nhtsa.gov/safecar/ConnectedVehicles/>
- [6] Drive C2X project, <http://www.drive-c2x.eu/project>
- [7] B. G. Jung, S. D. Jung, S. S. Lee, and H. O. Choi, "The design of interface about V2V/V2I based communication application service," in *Proc. KSAE 2008*, vol. 4, pp. 1760-1765, Daejeon, Korea, Apr. 2008.
- [8] H. T. Ju, D. M. Kim, Y. M. Jeong, and S. Y. Min, "A gateway platform for automotive synchronous ethernet network system," in *Proc. KSAE Conf.*, pp. 1177-1182, May 2013.
- [9] Autosar, <http://www.autosar.org/>
- [10] D. D. Park, H. H. Keum, J. Y. Park, and S. S. Hong, "Interconnecting AUTOSAR and MOST networks using an application-level

gateway component,” in *Proc. KSAE*, pp. 937-942, Jeju Island, Korea, Apr. 2009.

- [11] H. S. Seo, B. C. Kim, P. S. Park, C. D. Lee, and S. S. Lee, “Design and implementation of a UPn-CAN gateway for automotive environments,” *Int. J. Automotive Technology (IJAT)*, vol. 14, no. 1, pp. 91-100, Feb. 2013.
- [12] BOSCH, “CAN specification version 2.0,” 5 Apr. 1995, www.can-cia.org
- [13] J.-L. Bechennec, M. Briday, S. Faucou, and Y. Trinquet, “Trampoline an open source implementation of the OSEK/VDX RTOS specification,” in *Proc. ETFA*, pp. 20-22 Sept. 2006.
- [14] OSEK/VDX, “Open systems and the corresponding interfaces for automotive electronics,” <http://www.osek-vdx.org/>, 1995.
- [15] DSRC Implementation Guide, *A guide to users of SAE J2735 message sets over DSRC*, SAE International, 2010.

김 주 영 (Ju-young Kim)



2011년 8월 : 한양대학교 컴퓨터공학과 공학사
 2011년 9월~현재 : 한양대학교 전자컴퓨터통신공학과 공학석사과정
 <관심분야> V2X 통신, 차량 소프트웨어 플랫폼

서 현 수 (Hyun-soo Seo)



2005년 2월 : 부경대학교 제어계측공학과 공학사
 2007년 2월 : 부경대학교 제어계측공학과 공학석사
 2007년 12월~2010년 7월 : S&T 대우 기술연구소
 2010년 9월~현재 : 한양대학교 전자컴퓨터통신공학과 박사과정
 <관심분야> 차량 통신 기반 안전시스템, 차량 통신 및 네트워크, 커넥티드 카

이 상 선 (Sang-sun Lee)



1978년 2월 : 한양대학교 전자공학과 공학사
 1983년 2월 : 한양대학교 전자공학과 공학석사
 1990년 8월 : University of Florida 전기공학과 공학박사
 1991년 4월~1991년 11월 : 생산기술연구원 선임연구원 겸 조교수
 1991년 11월~1993년 2월 : 전자부품종합기술연구소 선임연구원
 1993년 3월~현재 : 한양대학교 융합전자공학부 정교수
 <관심분야> ITS 및 텔레매틱스, 통신망 통합 및 연동, 통신 네트워크, 위치추위 기술