Journal of Convergence for Information Technology
Vol. 9. No. 7, pp. 25-32, 2019
e-ISSN 2586-4440
DOI : https://doi.org/10.22156/CS4SMB.2019.9.7.025

리프로그래밍 시간 단축을 위한 차량 게이트웨이 개선 방안

김진호, 하경재*
경남대학교 컴퓨터공학부 교수

Method of In-Vehicle Gateway to Reduce the Reprogramming Time

Jin-Ho Kim, Kyung-Jae Ha*
Professor, Division of Computer Engineering, Kyungnam University

요 약 본 논문에서는 차량 ECU(Electronic Control Unit)의 리프로그래밍 시간 단축을 위한 차량 게이트웨이 개선 방안을 제안한다. 리프로그래밍 시간 단축을 위해 게이트웨이는 리프로그래밍을 진행하는 동안 리프로그래밍하는 ECU가 연결된 통신 채널에 리프로그래밍 이외의 메시지 전송을 금지하여야 하며, 이때 메시지 전송 금지로 인해 특정 ECU가 CAN 통신 수신 불가로 인한 고장이 발생하지 않도록 할 수 있어야 한다. 또한, 게이트웨이 내의 버퍼 오버플로우를 방지하기 위해 연속된 통신 메시지 전송 시 추가하는 지연 시간(STmin)을 최소로 할 수 있어야 한다. 이를 위해 본 논문에서는 UDS(Unified Diagnostic Services)의 링크 제어 명령 및 최신 MCU(Micro Controller Unit)에서 제공되는 HW 게이트웨이 기능을 이용한 개선 방안을 제안한다. 제안한 개선 방안은 차량에서 널리 사용되고 있는 인피니언사의 TC275 기반 임베디드시스템을 이용하여 구현하였으며, 개선된 실험 결과를 제시한다.

주제어 : 게이트웨이, 리프로그래밍, 캔, STmin, 차량 네트워크, 소프트웨어 업데이트

Abstract This paper proposes the method of an in-vehicle gateway to reduce the reprogramming time for the ECU (Electronic Control Unit). In order to reduce the reprogramming time, the gateway must prohibit transmitting messages, that are not related to reprogramming, to the destination CAN network, and no ECU should diagnose the DTC(Diagnostic Trouble Code) that indicates CAN communication error caused by prohibiting CAN messages by the gateway. Moreover, STmin, which are the minimum time between two consecutive CAN messages, should be minimized. In order to do this, this paper proposes the method that uses the link control command specified in UDS(Unified Diagnostic Services) and hardware based gateway functionality that are supported by the latest MCU(Micro Controller Unit). The proposed method is developed using TC275 based embedded system, and its results are presented.

Key Words : Gateway, Reprogramming, CAN, STmin, In-Vehicle Network, Software update

1. 서론

최근 자율 주행, 연비, 편의, 안전, 환경 규제 등 다양한 요구 사항을 만족하기 위해 차량에 장착되는 ECU(Electronic Control Unit) 개수가 급격하게 증가하고 있다. 뿐만 아니라 기존 ECU의 복잡도도 급격하게 증가하고 있어 기존 대비 양산 자동차에 소프트웨어 오류가 빈번하게 발생하고 있으며, 이를 해결하기 위해 차량 ECU 소프트웨어 업데이트 기능을 개발하여 사용하고 있다. 현재 차량에서 사용하고 있는 ECU는 장착된 MCU(Micro Controller Unit)의 특성에 따라 주행 중에 소프트웨어를 업데이트할 수 없으며, 소프트웨어를 업데이트하기 위해서는 차량 운행을 멈추고 업데이트가 완료될 때까지 대기하여야 한다. 최신 MCU에서는 주행 중 소

*Corresponding Author : Kyung-Jae Ha(kjha@kyungnam.ac.kr)
Received June 18, 2019
Revised July 5, 2019
Accepted July 20, 2019
Published July 28, 2019

프트웨어를 업데이트할 수 있는 기능을 제공하고 있으나, 보수적인 자동차 업계의 특성과 비용 상승을 고려할 때 대다수 차량 내 제어기는 기존 방식의 MCU를 사용할 것으로 전망된다.

현재 무선 통신을 이용하여 공식 서비스 센터에 방문하지 않고 차량 소프트웨어를 업데이트할 수 있는 OTA(On The Air)[1,2] 기술이 개발되었으나, 상기 설명한 사유로 주행 중 업데이트가 불가능하기 때문에 운전자는 소프트웨어 업데이트 도중 차량 운행을 할 수 없다는 단점이 있다. 뿐만 아니라 업데이트 시간이 길어질 경우 배터리 방전 등의 문제가 발생할 수 있어 차량에 심각한 문제를 야기할 수 있다. 따라서 현재 차량 소프트웨어 업데이트 시간을 단축하는 다양한 연구가 수행되고 있으며 자율 주행 자동차 등 복잡한 소프트웨어를 기반으로 동작하는 미래형 자동차에서는 소프트웨어 업데이트 시간 단축이 더 중요해질 전망이다.

다수의 ECU를 효율적으로 연결하기 위해 CAN(Controller Area Network), CAN FD(Flexible Data-Rate), FlexRay, Ethernet 등의 다양한 차량 네트워크(In-Vehicle Network)가 제안되었으며[3,4], 이와 같은 다양한 차량 네트워크를 연결하기 위해 차량 게이트웨이가 개발되었다. 현재의 차량 네트워크는 샤시, 파워트레인, 바디, 멀티미디어, ADAS 등의 도메인 별로 나뉘어져 있으며, 각 도메인 별 네트워크는 게이트웨이를 이용하여 연결된다.

차량 게이트웨이는 서로 다른 도메인 간 네트워크의 연결을 위해 반드시 필요한 장치로 다수의 ECU가 장착된 차종을 중심으로 최근 사용이 증가되고 있으나, 게이트웨이 장착 시 고속 데이터 전송이 불가하여 리프로그래밍 시간은 더욱 증가하는 단점이 있다. 기존의 게이트웨이가 적용되지 않은 차량의 경우 연속된 메시지 간에 지연 없이 CAN 메시지를 전송할 수 있었으나, 게이트웨이가 장착된 경우 게이트웨이 내에서 버퍼 오버플로우를 방지하기 위해 연속된 CAN 메시지 간에 지연시간을 추가해 주어야 한다. 연속된 CAN 메시지간의 지연 시간은 CAN TP(Transport Protocol)[5]에서 STmin이라 정의하고 있으며, STmin 값은 해당 차량에 장착된 게이트웨이에의 성능에 따라 결정된다. 일반적인 게이트웨이가 장착된 차량의 STmin 값은 수백 us 정도로 설정된다. 따라서, 하나의 CAN 메시지를 전송하는데 대략 250us가 소요되기 때문에, 수백 us의 STmin은 전송 시간을 2배 내지 3배 정도 증가시켜 차량 리프로그래밍 시간이 더 소요되게 하는 주된 원인이 된다.

기존의 리프로그래밍 시간을 단축시키는 연구로는 압축 리프로그래밍[6], Delta Flashing[7], 병렬 리프로그래밍 기법[8]이 제안되었다. 압축 리프로그래밍은 리프로그래밍 할 SW를 압축 알고리즘을 사용하여 용량을 축소하여 전송하고, 이를 수신한 ECU에서 압축을 해제하여 리프로그래밍을 진행하는 방식이다. Delta Flashing은 기존의 SW와 신규로 생성된 SW에서 서로 다른 점을 delta file로 생성하여 전송하고, delta file을 수신한 ECU는 저장된 기존 SW의 정보와 delta file을 병합하여 신규 SW를 생성한다. Delta Flashing은 압축 방식의 하나로, 기존 SW와 신규 SW의 정보를 이용하여 압축율을 월등히 향상 시켜 CAN으로 전송하는 데이터의 크기를 크게 감소시켜 전체 리프로그래밍 시간을 향상시킨다. 마지막 병렬 리프로그래밍 방법은 진단기와 게이트웨이가 Ethernet과 같은 고속의 통신 프로토콜을 이용하는 경우로서 여러 도메인에 연결된 제어기의 SW를 동시에 수신하여, 여러 도메인 네트워크를 동시에 리프로그래밍하는 방법이다. 이 방법은 여러 개의 도메인에 장착된 다수의 ECU를 리프로그래밍할 때 전체 리프로그래밍 시간을 단축할 수 있으나, 본 논문에서 다루는 단일 ECU를 리프로그래밍하는 시간 단축과는 다른 기술이다.

기존의 단일 ECU에 대한 리프로그래밍 속도 증가 기술은 주로 통신으로 전송하는 데이터의 사이즈를 줄이는 방향으로 연구되었다. 이런 연구는 실제 리프로그래밍에서 가장 큰 시간을 소요하는 CAN 전송 시간을 줄여 리프로그래밍 시간을 단축시키는 것이다. 하지만, 기존 연구에서는 본 논문에서 다루고자 하는 게이트웨이 버퍼 오버플로우 방지를 위해 STmin을 추가함으로써 발생하는 통신 지연 문제에 대해서는 고려하고 있지 않다. 따라서, 압축, Delta Flasing 방식 모두 게이트웨이가 장착된 차량에서는 기존 대비 더 오랜 리프로그래밍 시간이 소요된다.

본 논문에서 다루고 있는 게이트웨이에서의 지연 문제는 게이트웨이 내부 SW의 우선순위 등에 의해 CAN 메시지를 수신하여 처리하는 속도가 CAN 메시지 전송 속도보다 느린 경우, 게이트웨이 내의 수신 버퍼에 새로운 CAN 메시지가 기존의 처리되지 않은 CAN 메시지를 대체하여 기존 메시지가 손실되는 것을 방지하기 위해 연속적인 CAN 메시지 사이에 지연 시간 (STmin)을 추가함으로써 발생하는 문제이다. 이 문제를 해결하기 위해 FPGA를 이용한 게이트웨이[9]가 제안되었으나, 가격이 중요한 차량 산업의 특징상 FPGA와 같은 고가의 부품이 실제 차량 게이트웨이에 적용되기는 어렵다. 본 논문에서는 상기 문제를 개선하기 위해 인피니언사 TC275와 같은 MCU에서 제공하고 있는 HW

메시지 전달 기능을 이용한 개선 방법을 제안하고, 이를 실제 임베디드 시스템을 이용하여 개발한 후 개선된 결과를 제시한다.

2장은 논문의 이해를 돕기 위해 차량 게이트웨이, 리프로그래밍, CAN TP 기술을 간략히 설명한다. 3장에서는 본 논문에서 다루는 문제의 원인을 분석하고, 개선 방안을 제시한다. 4장에서는 임베디드 시스템을 이용하여 제안 방법을 구현하고 이에 따른 시험 결과를 제시한 후 5장에서 본 논문을 마무리한다.

2. Background

본 장에서는 제안 방안 설명의 이해를 돕기 위해 관련된 기술인 차량 네트워크 및 게이트웨이, 리프로그래밍, CAN TP 기술을 간략히 설명한다.

2.1 차량 전기/전자 아키텍처 및 게이트웨이

차량 내 ECU는 Fig. 1과 같이 파워트레인, 샤시, 바디, 멀티미디어 도메인으로 크게 나뉘어져 있으며, 서로 다른 도메인 간 정보 교환을 위해 중앙 게이트웨이[10]가 개발되었다. 서로 다른 도메인 간의 융복합 기능 개발을 쉽게 하기 위해 중앙 게이트웨이 기반의 아키텍처는 Fig. 2와 같은 도메인 제어 장치 기반 아키텍처로 2015년 이후 변경될 것으로 예상되었으나[11], 아직 차량에 적용되지 못하고 있는 실정이다.

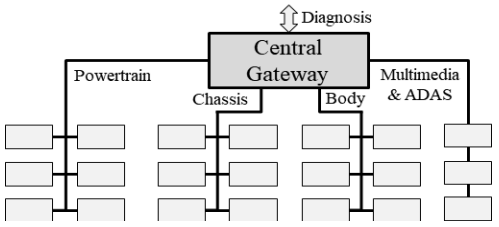

Fig. 1. Central gateway based E/E architecture

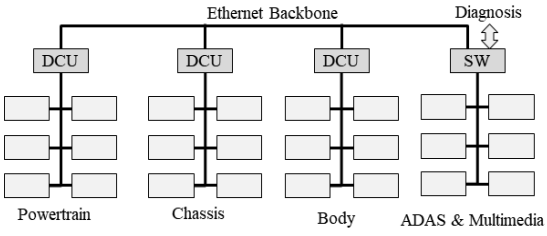

Fig. 2. Domain control unit based E/E architecture

Ethernet을 이용하여 게이트웨이와 진단기를 연결할 수 있는 DoIP(Diagnostic over Internet Protocol)도 개발되었으나, 가격 상승, 진단기 개발의 어려움, SAE J1979 법규 진단 통신 표준 미지원 등의 문제로 실제 양산 차량에 적용되지 못하고 있고, FlexRay와 같은 새로운 차량 통신 프로토콜은 가격 및 SW 변경량 등을 이유로 일부 고급 차량을 제외하고는 적용되지 못하고 있어 현재 대다수의 차량에서는 서로 다른 도메인의 CAN 네트워크를 연결해 주는 CAN-CAN 게이트웨이가 사용 중이다. 따라서, 차량 리프로그램시 외부 진단기는 게이트웨이로 CAN을 통해 업데이트할 SW를 전송하고, 게이트웨이가 업데이트할 ECU가 있는 도메인 네트워크로 메시지를 전달한다.

2.2 리프로그래밍 절차

차량 ECU에 탑재된 SW를 업데이트하는 방법을 리프로그래밍(Reprogramming)이라 부른다. 일반적으로 ECU는 탈착이 쉽지 않고, 보안 및 방수 등의 이유로 ECU의 케이스를 분리할 수 없기 때문에 통신을 이용한 SW 업데이트 방식을 사용한다. 모든 차량에는 법규 인증을 받을 때 사용하는 OBD(On Board Diagnostic) 커넥터가 운전석 왼쪽 아래에 장착되어 있으며, 이 커넥터에는 일반적으로 내부 ECU와 통신할 수 있는 CAN 통신 인터페이스를 제공한다.

기존에 게이트웨이가 장착되기 전에는 모든 도메인 네트워크의 CAN 통신을 OBD 단자에 연결하여, 외부 진단기가 직접 개별 ECU에 메시지를 송수신할 수 있었다. 게이트웨이가 장착되는 경우 보안[12,13]을 위해 각 도메인 네트워크는 OBD 단자에 연결되지 않고, 게이트웨이의 진단 채널만 OBD에 연결된다. 따라서 외부 진단기는 게이트웨이를 통해 각 도메인의 ECU와 데이터를 송수신한다. 리프로그래밍을 위한 세부 통신 방식은 차량 제조사마다 조금씩 다른 방식을 사용하지만 아래 Fig. 3와 같은 절차를 기반으로 동작한다.

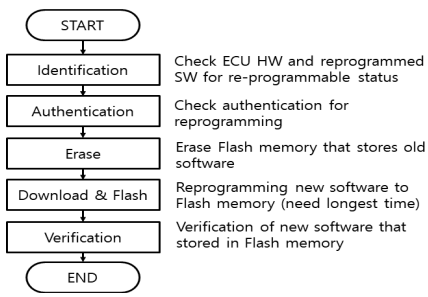

Fig. 3. Reprogramming sequence[11]

2.3 CAN TP

CAN은 한번에 최대 8byte의 데이터를 전송할 수 있기 때문에, SW 업데이트와 같이 대용량 데이터를 전송하기 위해서는 8byte 보다 큰 데이터를 송수신하기 위해 CAN TP를 사용한다. CAN TP는 Fig. 4.와 같이 8byte 보다 큰 데이터를 CAN으로 전송하기 위해 여러 개의 CAN 메시지로 분할하여 전송한다. CAN TP는 TCP의 Flow Control과 유사하게 수신 노드의 버퍼 오버플로우를 방지하기 위해 송신 메시지의 속도를 조정할 수 있어야 하며, 이를 위해 연속된 CAN 메시지 사이의 최소 시간 간격인 STmin을 사용한다.

게이트웨이가 미장착되어 있는 경우 STmin은 리프로그래밍이 필요한 ECU의 성능에 의해 결정되며, 일반적으로 지연 시간이 없는 0으로 설정된다. 게이트웨이가 장착되는 경우 모든 메시지는 게이트웨이를 통하여 ECU에 전달되기 때문에 게이트웨이의 오버 플로우 방지를 위해 STmin 값이 추가된다.

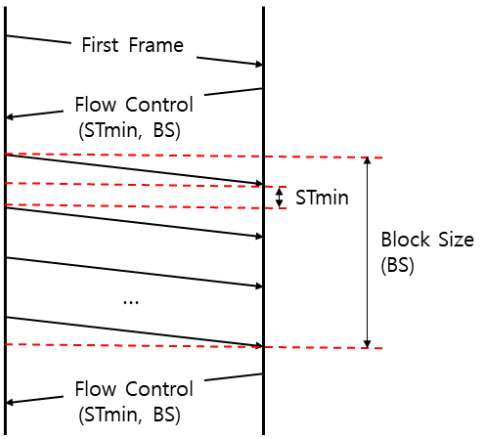

Fig. 4. Consecutive message transmission of Can TP

3. 문제 분석 및 개선 방안

본 절에서는 게이트웨이에 의해 리프로그래밍 시간이 증가할 수 있는 원인을 분석하고 이를 개선할 수 있는 방법을 제안한다.

3.1. 게이트웨이 CAN 전송 지연 원인 분석

게이트웨이가 장착된 차량에서 CAN 전송 지연은 CAN 네트워크에 리프로그래밍과 관계없는 다른 메시지가 전송되거나, 앞서 설명한 STmin 설정에 따른 CAN의 대역폭 중 일부가 리프로그래밍과 관계없는 메시지의 전송 또는 연속 메시지 간의 지연 시간(STmin)으로 낭비되기 때문에 발생한다. 이와 같은 게이트웨이 대역폭 손실 문제는 3가지 원인이 있다.

첫 번째 경우는 게이트웨이가 수신한 메시지를 목적 네트워크로 전달하려고 할 때, 목적 네트워크에 이미 다른 CAN 메시지가 전송 중인 경우이다. 이 문제는 기존의 게이트웨이가 장착되기 전의 차량에서도 발생할 수 있는 문제로 이를 방지하기 위해 진단 통신 표준인 UDS[14]에 통신 금지 명령이 제공된다. 통신 금지 명령을 수신한 제어기는 진단 통신 이외의 모든 CAN 통신 송신을 중단하며, 진단 통신은 진단기와 1:1로 연결되는 통신이므로 리프로그래밍을 하고 있는 ECU만 실제 CAN 메시지를 송수신하게 된다. 기존 게이트웨이가 미장착된 차량에서는 UDS의 통신 금지 명령을 받아도 CAN 메시지를 전송하는 ECU가 있는 경우, CAN의 우선 순위 경쟁에 의해 리프로그래밍 시간이 조금 더 소요되는 것 이외에는 문제가 없었으나, 게이트웨이가 장착되는 경우 메시지 손실 문제가 발생할 수 있어 모든 ECU는 UDS 메시지 송신 중단 명령을 UDS 표준에 맞게 개발해야 한다.

두 번째 경우는 게이트웨이를 통해 다른 도메인 네트워크의 메시지가 리프로그래밍하려는 ECU가 장착된 네트워크로 전달되거나, 게이트웨이가 송신자의 메시지가 리프로그래밍 하려는 ECU가 장착된 네트워크로 전송할 때 발생할 수 있다. 다른 네트워크의 메시지가 전달되는 경우는 리프로그래밍 진입 시 진단기가 송신하는 UDS의 통신 전송 금지 명령을 게이트웨이가 모든 도메인의 네트워크로 전달하여, 모든 채널의 통신 메시지가 전송되지 않도록 하여 해결할 수 있다. 또한, 게이트웨이도 UDS의 통신 전송 금지 명령을 수신하면 진단 통신 메시지를 제외한 모든 메시지 송신을 금지하여야 한다. 차량에서 CAN 통신 메시지는 매우 중요하기 때문에 대다수의 ECU는 CAN 메시지의 정상 수신 여부를 확인하여 문제가 발생하면 고장 여부를 저장하고, 운전자에게 MIL(Malfunction Indicator Lamp)을 점등하여 알려준다. 따라서 UDS의 통신 전송 금지 명령을 수신하는 모든 ECU는 CAN 메시지 전송 금지뿐 아니라, CAN 고장 진단 기능을 비활성화하여, CAN 메시지가 미수신되어 고장이 잘못 진단되지 않도록 방지해야 한다.

세 번째 원인은 Fig. 5와 같이 게이트웨이 내부 SW에 의해 소스 네트워크의 메시지를 목적지 네트워크로 전달하는 것이 지연되는 경우이다. Fig. 5는 설명의 편의를 위해 FIFO를 사용하지 않는 경우로 게이트웨이 내에서 T_d시간

만큼 지연된 후 MSG#2가 전달되어 목적 네트워크에 T_d시간 동안 아무런 CAN 메시지를 전송하지 않고 낭비된다.

이와 같은 문제를 개선하기 위해 FPGA를 이용한 HW 기반 게이트웨이[9]가 제안되었으나, 가격이 중요한 자동차 산업에 고가인 FPGA가 활용되기는 어렵다. 따라서, 대부분의 차량 게이트웨이는 MCU 기반의 SW로 개발되어, Fig. 5에서와 같은 SW에 의한 지연 문제는 불가피하게 발생하게 된다.

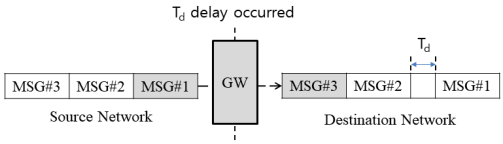

Fig. 5. Description of gateway delay

상기 설명한 게이트웨이에서의 지연 문제는 단순히 리프로그래밍 시간이 오래 걸리는 문제가 아니라, 일부 CAN 메시지가 손실되어 리프로그래밍 신뢰성에 문제를 일으킬 수 있다. Fig. 6는 설명의 편의를 위해 FIFO를 사용하지 않는 단일 버퍼를 사용하는 게이트웨이의 예로서, SW 지연이 발생하여 수신 버퍼의 메시지를 목적 버퍼로 전달하지 못하고 있는 도중에 새로운 메시지가 수신되는 경우(3번 메시지)이다. 이 경우에는 기존의 수신 버퍼에 있던 메시지(2번 메시지)가 손실되고, 새로운 메시지가 SW 지연 시간(T_d) 이후에 목적 네트워크 전송용 버퍼로 전달되어 송신된다. 신뢰성이 중요한 차량에서 CAN 메시지의 손실은 반드시 개선되어야 한다. 이 문제를 해결하기 위해 게이트웨이가 장착된 차량에서는 CAN TP로 연속된 메시지를 전송할 때 게이트웨이의 최대 지연 시간(T_d)을 고려하여, 연속 메시지 사이의 최소 시간(STmin)을 설정한다. 게이트웨이 최대 지연 시간은 게이트웨이의 성능에 따라 다를 수 있으나, 국내 제조사의 경우 수백 us 정도의 지연이 필요하다. 일반적으로 500Kbps로 8byte 메시지를 전송하는데 약 190 ~250us 정도 소요되기 때문에 수백 us의 STmin은 전체 CAN 전송 시간을 2배 이상 증가시키는 문제를 야기한다.

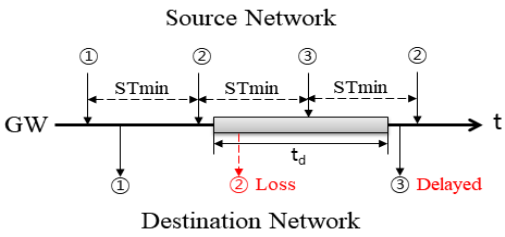

Fig. 6. Message loss problem caused gateway delay

3.2. 개선 방안

게이트웨이에서 메시지 전송이 지연되는 3가지 원인 중 첫 번째, 두 번째 문제는 UDS의 메시지 전송 금지 명령을 게이트웨이가 특정 채널이 아닌 모든 채널로 전송하고, 게이트웨이 자신도 진단 통신 이외의 메시지를 송신 및 라우팅을 금지하여 간단하게 해결할 수 있으므로, 추가적인 설명은 생략한다.

세 번째 경우는 앞서 설명한 바와 같이 게이트웨이 내에서 리프로그래밍에 사용하는 메시지 전달이 지연되어 발생하므로, 리프로그래밍 시 리프로그램 메시지 전달 SW가 지연되지 않도록 개선하면 된다. 본 논문에서 수행한 실험(4장 참조)에 의하면 500Kbps CAN 통신 사용 시에 200us 정도의 게이트웨이 내의 지연은 메시지 손실을 일으키지 않으므로, 연속된 메시지 사이의 지연 시간 추가 없이 리프로그래밍을 진행할 수 있다. 따라서 가장 간단한 해결 방법은 리프로그래밍을 진행하는 시점에 리프로그래밍용 SW의 우선순위를 높게 변경하여, CAN 메시지의 최대 지연이 200us 이내가 되도록 개발하는 것이다. 하지만, 실제 차량 게이트웨이는 메시지 전달 이외에 네트워크 관리, 전원 관리, 진단 등 다양한 기능을 수행하고 있으며, 각 기능의 우선순위는 중요도에 따라 신중하게 결정되므로 리프로그래밍을 위해 간단히 변경하기는 어렵다. 우선순위 변경은 게이트웨이 상에 동작하는 모든 SW에 영향을 주어 차량 신뢰성에 문제를 야기 할 수 있다[15].

앞서 설명한 우선순위를 높여 SW 지연을 최소화하는 방법은 최근 사용이 증가하고 있는 멀티코어 기술을 사용하면 해결할 수 있다. 멀티코어 MCU를 사용하는 경우 특정 코어에 리프로그래밍 시 기능이 중지 또는 지연되어도 되는 기능만 할당하고, 리프로그래밍 시에도 정상적으로 동작해야 하는 기능은 다른 코어에 할당한다. 멀티코어 활용 시 SW 실행을 담당하는 인터럽트 및 태스크의 우선순위는 해당 코어에서만 적용되기 때문에, 리프로그래밍 시에 특정 인터럽트나 태스크의 우선순위를 높게 설정해도 다른 코어에는 영향을 미치지 않는다. 다만, 이 방법은 기존의 한 개의 코어에서 수행되던 게이트웨이 기능을 두 개 이상의 코어에서 활용해야 하며, 멀티코어 기반의 게이트웨이 개발이 필요하여 실제 차량에 적용될 확률은 크지 않다.

다른 방법으로는 FIFO(First In First Out) 구조의 송/수신 버퍼를 사용하는 것이다. 충분한 버퍼 크기를 갖는 FIFO를 사용하면 일시적으로 발생하는 SW 지연에 의한 메시지 손실이나 전송 지연 문제를 방지할 수 있다. 다만, 이

방법은 일시적인 SW 지연에 대한 문제를 개선할 수 있으나 긴 지연이 발생하는 경우 동일한 문제가 발생한다. 따라서, 이 방법을 활용하면 메시지 손실율을 줄일 수는 있으나 완전한 해결이 불가능하여, 기존의 해결 방법인 연속된 CAN 메시지 사이의 지연 시간을 추가해야 한다.

또 다른 해결 방법으로는 게이트웨이 내에서 수신 버퍼에 메시지가 수신 속도와 송신 버퍼로 메시지를 송신하는 속도를 비교하여 수신 속도가 빠른 경우 STmin을 증가시켜, 실행 중 메시지 누락 가능 여부를 확인하여 동적으로 STmin 값을 가변시키는 것이다. 이 방법은 메시지 수신 버퍼로서 FIFO를 사용하고, 수신 인터럽트 발생 시 2개 이상의 메시지가 수신되어 있다면 SW 지연으로 인해 수신 인터럽트 발생이 누락된 것으로 간주하고 STmin을 증가시킨다. 반대로 목적 네트워크의 전송 완료 인터럽트 수행 시 수신 버퍼에 저장된 메시지 수가 일정 수보다 적으면 STmin을 다시 감소시킨다. 이 방법은 일반적인 상황에서 메시지 누락 문제를 해결할 수 있다. 그러나 게이트웨이 내에서 CPU 사용율이 급격히 증가하여 긴 시간 SW 지연이 발생하는 시점에 연속으로 CAN 메시지가 수신되어 수신 FIFO에 버퍼 오버플로우가 발생하는 등의 최악의 조건에서는 여전히 메시지 누락 문제가 발생할 가능성이 있으며, 안전에 대한 우려로 인한 보수적인 알고리즘 개발은 STmin 가파른 증가와 느린 감소로 인해 기존의 수백 us 수준의 고정 STmin을 설정하는 방법보다 더 오랜 시간이 소요될 가능성도 존재한다.

앞서 설명한 바와 같이 게이트웨이 내에서의 SW 지연 문제는 SW적으로 해결하기 매우 어렵다. 따라서 이 문제는 HW로 해결하는 것이 바람직하다. 앞서 설명한대로 비용 문제로 차량 게이트웨이를 FPGA를 이용하여 개발할 수 없기 때문에 이 문제를 해결하기 위해 주요 차량 MCU 업체의 최신 CAN Controller에는 하드웨어 게이트웨이 기능이 포함되어 있다. 이 기능은 수신 버퍼를 게이트웨이 모드로 사용하겠다고 설정하게 되면, 수신 버퍼에 메시지가 수신될 때 CAN Controller로 하여금 수신 버퍼의 메시지를 목적 버퍼로 복사 후 목적 네트워크로 전송하게 한다. 이 방법은 기존 게이트웨이가 사용하는 MCU 회사별로 최신 MCU에서 제공하고 있는 기능이므로, 기존 게이트웨이 SW를 그대로 사용하면서 간단한 레지스터 변경만으로 개발할 수 있으며, HW를 통한 메시지 변환을 통해 SW에 의한 지연 문제를 원천적으로 개선할 수 있다.

4. 구현 및 실험 결과

본 장에서는 앞서 설명한 게이트웨이의 SW 지연에 따른 메시지 손실 문제를 실험을 통해 증명하고, 제시한 HW 메시지 포워딩 방법을 사용하여 문제를 개선한 결과를 제시한다.

4.1. 실험 환경

Fig. 7은 본 논문의 실험을 위해 사용한 실험 환경을 나타낸다. 게이트웨이는 소스 네트워크에 연결된 진단기로부터의 CAN 메시지를 수신하여, 목적 네트워크인 CAN 네트워크로 전달한다. 전달된 메시지는 메시지의 손실 여부를 분석하기 위해 Kvaser사의 Leaf Light를 이용하여 PC에 연결된다. PC에서는 Kvaser사에서 제공하는 CAN 모니터링 프로그램인 CAN King을 사용하여 게이트웨이가 전달한 메시지의 손실 여부를 확인하였다.

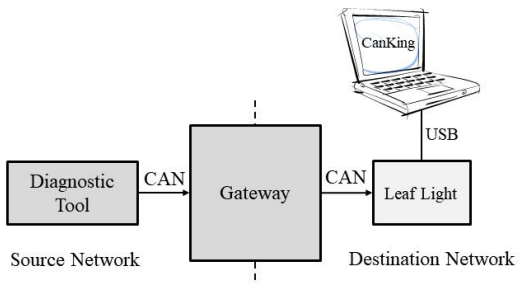

Fig. 7. Experimental environment

게이트웨이는 MCU에 내장된 CAN Controller에 게이트웨이 기능이 탑재된 인피니언사의 TC275를 사용한 임베디드 시스템을 사용하였다. 실제 차량 환경에서의 진단기는 전용 장비 또는 PC를 사용하나, 본 논문에서는 정밀한 STmin 값 설정을 위해 PC보다는 게이트웨이와 동일한 임베디드 시스템을 사용하였다. Fig. 8은 본 논문에서 진단기 및 게이트웨이 개발에 이용한 임베디드 시스템의 사진이다.

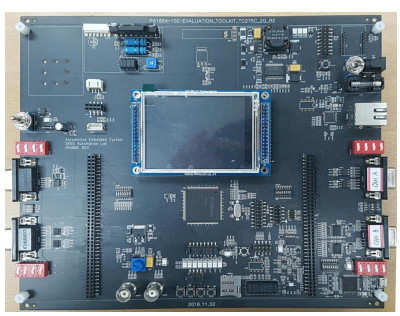

Fig. 8. TC275 based embedded system

4.2. 구현 및 실험 결과

실험을 위한 게이트웨이는 간단히 진단기가 연결된 소스 네트워크로부터 CAN 메시지가 수신되면 인터럽트를 발생하여, 인터럽트 내에서 목적 네트워크로 수신한 CAN 메시지를 전송한다. 게이트웨이에 의한 SW 지연 시간에 따른 메시지 손실율을 측정하기 위해, 타이머를 이용하여 1ms마다 인터럽트를 발생시켜 300us, 250us, 200us가 지연되도록 설정하였다. 이때 1ms 마다 발생하는 인터럽트는 CAN 수신 인터럽트보다 우선순위를 높게 설정하여, CAN 메시지 전달 작업이 설정된 시간만큼 지연될 수 있도록 설정하였다. 실험을 단순화하기 위해 진단기는 매번 1씩 증가하는 카운트 값을 가진 80,000개의 CAN 메시지를 STmin 값을 0, 300us로 설정하여 전송한 후 목적 네트워크로 전달된 CAN 메시지를 CAN King으로 분석하여 손실된 메시지의 개수를 측정하였다. Table 1은 80,000개 메시지 중에 손실된 메시지의 개수와 손실율을 보여준다. 아래 결과를 통해 SW 기반으로 개발된 게이트웨이에서 SW에서 발생할 수 있는 최대 지연 시간보다 큰 STmin을 설정해야 메시지 손실 없이 통신이 가능하다는 것을 확인할 수 있다.

Table 1. Data loss rate caused by SW delay

STmin \ SW delay	300us	250us	200us
0	10399 (13.0%)	2196 (2.7%)	0
300	80 (0.001%)	0	0

본 논문에서 제안하는 HW 기반 메시지 전달은 TC275에 포함되어 있는 CAN Controller인 MultiCAN+ 모듈의 메시지 수신 버퍼와 목적지 송신 버퍼의 설정을 변경해 주면 된다. 자세한 레지스터 설정 방법은 TC275 매뉴얼을 참조하면 된다. 게이트웨이 모드의 설정 후 수신 버퍼에 메시지가 수신되면 MultiCAN+ 모듈에 의해 자동으로 목적 버퍼로 복사된 후 목적 네트워크로 메시지가 전송된다. 따라서, SW의 지연 시간은 CAN 메시지 지연에 영향을 주지 않는다. 다만, 이 방법은 수신한 메시지 버퍼를 복사하는 방식이기 때문에 수신한 메시지의 ID, 데이터 등의 속성을 변경하지 못하며, PDU 라우팅이나 시그널 라우팅[10]과 같은 라우팅 방법은 지원하지 않지만, 리프로그래밍 시에는 해당 라우팅 방법은 필요하지 않다.

Table 2는 제안하는 HW 기반 메시지 전달 방법을 사용하여 1ms 마다 300us, 250us, 200us의 SW 지연을 추가하였을 때 메시지 손실율을 측정한 것이다. Table 1과 동일하게 80,000개의 메시지를 실험하였다. 시험 결과 제안 방법을 사용하면 SW 지연에 무관하게 STmin 0으로 설정하여 리프로그래밍 시간 지연 문제를 해결할 수 있다는 것을 확인하였다.

Table 2. Data loss rate using proposed method

STmin \ SW delay	300us	250us	200us
0	0	0	0
300	0	0	0

5. 결론

본 논문에서는 게이트웨이가 적용되는 경우 차량 리프로그램 시간이 증가하는 문제의 원인을 분석하고, 이를 개선할 수 있는 방법을 제안하였다. 제안한 방법은 최신 차량용 MCU에 적용되는 HW 기반 메시지 전달 방법을 사용하여, 게이트웨이 내의 SW 지연에 영향을 받지 않고 메시지 전달이 가능하다. 실험 결과 기존에 1ms마다 300us 정도의 SW 지연이 있는 게이트웨이에서 13% 정도의 메시지가 손실되었으나, 제안하는 HW 기반 메시지 전달 방법을 사용하는 경우 연속된 CAN 메시지 간의 지연 시간 없이 송신하는 경우에도 메시지 손실 없이 전송할 수 있어, 게이트웨이가 장착된 차량에서의 리프로그램 시간 증가 문제를 개선할 수 있다. 제안 방법은 FPGA와 같은 고가의 장비를 사용하지 않고, 멀티코어 기술 적용과 같은 SW의 대폭적인 변경 없이 간단한 레지스터 수정으로 적용이 가능하여, 향후 차량 게이트웨이에 적용할 수 있을 것으로 예상된다.

REFERENCES

[1] D. Mckenna. (2016). *Making Full Vehicle OTA Updates a Reality*, NXP Semiconductors (Online). https://www.nxp.com/docs/en/white-paper/Making-Full-Vehicle-OTA-Updates-Reality-WP.pdf

[2] Z. Fox. *A New Concept In OTA Updating For Automotive*. Auroralabs(Online). https://www.auroralabs.com/wp-content/uploads/2018/11/A-new-concept-in-OTA-Updating-for-Automotive-White-paper-web.pdf

[3] S. Hong. (2018). Research on Countermeasures of Controller Area Network Vulnerability. *Journal of Convergence for Infomation Technology, 8(5),* 115-120.
DOI : 10.22156/CS4SMB.2018.8.5.115

[4] K. J. Lee & K. H. Lee. (2013). Authentication Scheme using Biometrics in Intel. Vehicle Network. *Journal of the Korea Convergence Society, 4(3),* 15-20.
DOI : 10.15207/JKCS.2013.4.3.015

[5] ISO 15765-2. (2016). *Road vehicles -- Diagnostic communication over Controller Area Network (DoCAN) -- Part 2: Transport protocol and network layer services.*

[6] K. Ryozo, M. Satoshi, M. Mitsuhiro, N. Masayuki & K. Satoshi. (2009). A New Method of Fast Compression of Program Code for OTA Updates in Consumer Devices. *IEEE Transactions on Consumer Electronics, 55(2),* 812-817.
DOI : 10.1109/TCE.2009.5174459

[7] D. Bogdan, R. Bogdan & M. P. Work. (2016). Delta Flashing of an ECU in the Automotive Industry. *11th IEEE International Symposium on Applied Computational Intelligence and Informatics.* 503-508.

[8] Y. S. Lee, J. H. Kim, H. V. Hong & J. W. Jeon. (2015). A Parallel Re-programming Method for In-vehicle Gateway to save software update time. *IEEE International Conference on Information and Automation,* 1497-1502.

[9] S. Shreejith, P. Mundhenk, A. Ettner, S. A. Fahmy, S. Steinhorst, M. Lukasiewycz & S. Chakraborty. (2017). VEGa: A High Performance Vehicular Ethernet Gateway on Hybrid FPGA. *IEEE Transactions on Computers, 66(10),* 1790-1803.
DOI : 10.1109/TC.2017.2700277

[10] J. H. Kim, S. H. Seo, N. T. Hai, B. M. Cheon, Y. S. Lee & J. W. Jeon. (2015). Gateway Framework for In-Vehicle Networks based on CAN, FlexRay and Ethernet. *IEEE Transactions on Vehicular Technology, 64(10),* 4472-4486.
DOI : 10.1109/TVT.2014.2371470

[11] R. Schmidgall. (2012). *Automotive embedded systems software reprogramming.* Doctoral dissertation. Brunel University, London.
DOI : 10.22156/CS4SMB.2018.8.5.115

[12] M. Han & W. S. Bae. (2014). Security Verification of a Communication Authentication Protocol in Vehicular Security System. *Journal of Digital Convergence, 12(8),* 229-234.
DOI : 10.14400/JDC.2014.12.8.229

[13] S. J. Lee & W. S. Bae. (2015). Verification of a Communication Method Secure against Attacks Using Convergence Hash Functions in Inter-vehicular Secure Communication. *Journal of Digital Convergence, 13(9),* 297-302.
DOI : 10.14400/JDC.2015.13.9.297

[14] ISO 14229-3. (2012). *Road vehicles — Unified diagnostic services (UDS) — Part 3: Unified diagnostic services on CAN implementation (UDSonCAN).*

[15] Q. Zhu, H. Zeng, W. Zheng, M. D. Natale & A. S. Vincentelli. (2012). Optimization of Task Allocation and Priority Assignment in Hard Real-Time Distributed Systems. *ACM Transactions on Embedded Computing Systems, 11(4),* Article 85.
DOI : 10.1145/2362336.2362352

김 진 호(Jin-Ho Kim) [정회원]

· 2007년 2월 : 성균관대학교 컴퓨터공학과(공학사)
· 2009년 2월 : 성균관대학교 전기전자컴퓨터공학부 (공학석사)
· 2015년 2월 : 성균관대학교 전기전자컴퓨터공학부 (공학박사)
· 2015년 4월 ~ 2019년 2월 : 현대자동차 책임연구원
· 2019년 3월 ~ 현재 : 경남대학교 컴퓨터공학부 조교수
· 관심분야 : 차량 임베디드시스템 및 실시간 네트워크
· E-Mail : kimjh@kyungnam.ac.kr

하 경 재(Kyung-Jae Ha) [정회원]

· 1980년 2월 : 성균관대학교 전기공학과(공학사)
· 1982년 2월 : 성균관대학교 컴퓨터공학(공학석사)
· 1989. 2월 : 성균관대학교 컴퓨터공학(공학박사)
· 1980년 9월 ~ 1984년 3월 : 산업연구원 연구원
· 1997년 1월 ~ 1998년 1월 : 미국 웨인주립대 visiting scholar
· 1984년 3월 ~ 현재 : 경남대학교 컴퓨터공학부 교수
· 관심분야 : 컴퓨터 구조
· E-Mail : kjha@kyungnam.ac.kr