

자동차 네트워크 시스템의 효율적 관리를 위한 CAN의 동적 선행대기 열 기법

A Dynamic Precedence Queue Mechanism of CAN for An Efficient Management of Automobile Network System

최 호 식, 최 효 식, 이 장 명*
(Ho-Seek Choi, Hyo-Seek Choi, and Jang-Myung Lee)

Abstract : Today the automobile has been changed from a mechanical system to an electronic control system by the development of the electronic technology. In the automobile body, most of these electronic control devices are networked and managed fully by the CAN protocol. But, when a network system is overloaded, unexpected transmission delay for relative low priority objects occurs due to the static priority definition of the CAN protocol. To resolve this problem, this paper proposes a dynamic precedence queue mechanism that creates a queue for the low priority object and its relevant objects to be transmitted, which becomes urgent in an overloaded network system to keep the maximum allowable time delay. For the generated queue, the highest priority is assigned to transmit the queued objects within the shortest time. The mechanism is implemented in the logical link layer of CAN, which does not require any modification of the old CAN hardware. Effectiveness of the proposed mechanism is verified by the real experiments with an automobile network system.

Keywords : the automobile network system, CAN (Controller Area Network), data link layer

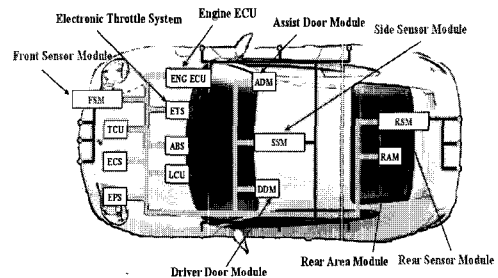
I. 서론

최근 자동화된 공장이나 전자화된 자동차 등 실시간 분산 제어가 요구되는 시스템들의 효율적 관리를 위하여 네트워크를 도입하고 있다. 특히, 자동차 산업은 관련 산업과의 기술적 진전에 힘입어 비약적인 발전을 거듭하여 오늘날 대중화와 더불어 신뢰성 및 안전성의 확보, 고급화, 운전의 편리성 등과 같은 기능의 구현에 대한 추구가 가속화 되고 있다. Throttle valve, brake system, engine과 같은 기존의 기계적인 시스템들이 새롭게 개발된 전자 unit들에 의해서 전자 제어 시스템화가 급속히 이루어 지고 있다. 또한, 자동차 산업의 고급화로 인하여 차내 전자 unit의 수가 증가하고, 이들을 통합한 시스템 통합 기법이 활발히 진행 중에 있다[1,2].

자동차 전자제어 시스템은 main system인 powertrain, chassis, body로 크게 3부분으로 분류된다. Powertrain system은 차량의 동력원 및 전달부분으로 엔진, 구동계, 보기류로 구성된다. ENG ECU(Engine ECU), Electronic Throttle System (ETS) 그리고 Transmission Control Unit (TCU)와 같은 engine 과 transmission control unit은 자동차의 가장 기본이 되는 제어장치이며 운전자가 운행 중 없어서는 안될 핵심적인 전자 제어장치이다. Chassis 분야는 차량주행 시 자세제어에 관련되는 기술로 크게 제동, 현가, 조향 system으로 구분된다. 그 중 제동 system의 대표적인 예는 현재 많이 보급되고 있는 Anti-lock Brake System(ABS)으로 미끄럼 방지뿐 아니라 제동 시 방향의 안전성, 즉 최적의 제동성과 조타성을 유지하는 효과를 나타낸

다. Body 분야는 주로 소비자의 다양한 요구 사항에 관련된 상품성 향상에 있다. Automatic door mirror system, auto door lock system, auto light control system, driving warning system 그리고 GPS을 이용한 navigation system과 같은 기능의 고급화에 따라 탑승자의 안락성, 편의성 및 안전기술에 중점을 두고 있다.

이러한 자동차 전자제어 unit들의 개발과 더불어 각각의 제어 장치들간의 효율적인 데이터 교환 및 전체 시스템을 통합 관리하기 위한 네트워크 시스템의 구축이 필요하다. 네트워크 시스템 중 자동차 네트워크 시스템에 널리 사용되고 있는 CAN 프로토콜은 차량 내부의 복잡한 배선 문제 및 이로 인한 신뢰도 저하를 해결하기 위해서 개발되었다[3,4]. CAN 프로토콜의 가장 큰 이점은 Intel사의 82527 CAN 제어기와 같은 전자 부품들의 가격이 저렴하다는 것이다. 사용하기 간편하고 가격이 저렴하다는 것은 자동화 네트워크를 구축함



TCU : Transmission Control Unit
ECS : Electronic Control Suspension
EPS : Electronic Power Steering
ABS : Anti-Lock Brake system
LCU : Light Control Unit

그림 1. 자동차의 CAN Network system 구성.
Fig. 1. Architecture of CAN Network system in automobile.

* 책임저자(Corresponding Author)

논문접수 : 2005. 6. 30., 채택확정 : 2006. 2. 8.

최호식, 최효식, 이장명 : 부산대학교 전자공학과

(hschoi@pusan.ac.kr/chs3040@pusan.ac.kr/jmlee@pusan.ac.kr)

※ 본 연구는 교육부에서 주관하는 "차세대물류IT기술연구사업단"에 의해 지원 받은 연구임.

에 있어서 가장 우선적으로 고려되어야 할 사항들이기 때문에 CAN의 보급이 지속되고 있다[5-7]. 또한, CAN은 메시지의 전송에 priority를 부여할 수 있어 산업현장에서 필요로 하는 긴급한 통신조건을 만족시킬 수 있다. 그리고 부품과 통신 보드가 다른 종류의 네트워크보다 저렴하며 네트워크 관리 방법도 편리하고 통신 소프트웨어도 용이하게 작성할 수 있다.

CAN에서 사용하는 CSMA/CD[8] 매체접근 기술은 충돌이 발생했을 때, 신호들을 파괴시키지 않고 충돌을 조정하는 조정자가 priority가 가장 높은 프레임을 전송하고, 나머지는 전송을 중지시킨다. 전송되는 프레임은 특별한 목적지를 가지고 있지 않으며 전체 네트워크 상에서 메시지에 따라 유일한 ID를 부여 받는 목적코드를 가진다. 이러한 ID는 네트워크 상에서 전송되는 다른 종류의 메시지 혹은 unit에 대해 priority를 절대적으로 할당한다. 이러한 기법은 네트워크 상에서 충돌이 일어났을 때 충돌을 조정하는 좋은 방법이 된다. 한 예로 자동차 네트워크 시스템에서 Engine ECU (ENG ECU), Electronic Throttle System (ETS)와 같이 자동차의 engine부분과 관련된 unit은 상대적으로 높은 priority를 가지고 있으며 자동차 주위 장애물 유무를 인식하고 warning alarm을 알리는 sensor module은 전체 네트워크 시스템 중 가장 낮은 priority로 설정된다.

차량이 도로를 정상주행 중일 때는 네트워크 상에 연결된 각각의 unit들이 원하는 시간 이내에 데이터를 송수신 할 수 있다. 그러나 차량의 주위환경 변화 및 고장으로 인하여 네트워크 상에 데이터 전송량이 급격히 증가하게 되면 priority가 상대적으로 높은 ENG ECU, ETS는 원하는 시간 이내에 데이터를 전송할 수 있지만 sensor module과 같은 상대적으로 낮은 priority를 갖는 unit은 충돌에 의해서 전송기회를 잃어버린다. 따라서, 원하는 시간 이내에 데이터를 전송하지 못하고 지연시간[9-11]이 발생하여 전체 시스템이 위험한 상황에 빠질 수 있다. 특히 차량의 저속 운행 및 주차 시 차량 외부 장애물 및 근접 차량의 거리정보 데이터가 증가하여 전체 네트워크 시스템의 부하가 증가하게 된다. 그러나 낮은 priority로 인해서 sensor module은 차량의 외부 장애물 거리 정보를 실시간으로 전송할 수 없게 되고, 주기적으로 warning alarm을 운전자에게 전달할 수가 없다. 최악의 경우 차량의 충돌이 발생하여 운전자 및 차량에 심각한 충격이 발생할 수도 있다.

따라서 본 논문에서는 과부하 시 CAN 네트워크 상에 정적으로 할당되는 ID로 인하여 데이터 전송의 비효율성을 제거하고 최대 허용 지연시간을 보장하기 위해서 네트워크 상에 과부하가 발생할 때 priority가 상대적으로 낮고 전송되는 데이터의 목적이 비슷한 unit들에 대해 각각 priority를 동적으로 재설정하고 대기 열을 생성하고 난 후 선행방식에 의해서 데이터를 전송함으로써 상대적으로 낮은 priority를 갖는 unit의 전송기회를 보장하고 지연시간을 최대한으로 줄일 수 있는 기법을 제시하였다[12].

이러한 선행대기 열은 정적으로 할당하는 것이 아니라 네트워크 시스템의 전송부하 조건에 따라 동적으로 할당되어 네트워크 상의 전송 효율을 높일 수 있다. 그리고 각각의 대기 열들은 상대적인 priority를 가지고 있으며 서로 독립적으

로 데이터 전송순서를 할당 할 수 있다.

차량의 주위환경 변화 및 고장으로 인하여 네트워크의 과부하가 발생하여 priority가 낮은 unit의 전송이 지속적으로 지연되는 경우에 이 unit의 priority를 동적으로 조정하여 일정시간 이내에 전송이 가능하도록 하는 기법을 확장 CAN 프로토콜을 기반으로 개발하고자 한다.

Engine ECU (ENG ECU), Electronic Throttle System (ETS)과 같이 상대적으로 priority가 높은 unit이 전체 네트워크를 독점하는 것을 방지하고, 최대 허용 전송 지연시간을 넘기지 않도록 모든 unit을 관리하기 위한 방법을 제시하는 것으로, 3계층으로 나누어진 CAN의 중간 계층인 데이터 연결 계층(data link layer)에서 실현된다. 이러한 변환은 CAN의 MAC 부계층(sub-layer)에 의해서 제공되는 부분과 서로 호환이 가능하기 때문에 표준 CAN 프로토콜에서 발전된 산업적인 다른 응용 분야에도 동일하게 적용할 수 있다. 즉, 본 논문에서 제시된 기법은 CAN 기반의 모든 응용기법과 호환적으로 사용될 수 있다.

본 논문의 2장에서는 CAN 프로토콜의 기본적인 이론을 서술하고 3장에서는 본 논문에서 제시하는 동적 선행대기 열 기법에 대해서 서술한다. 그리고, 4장에서는 실제 자동차 네트워크 시스템에 구축되어 있는 CAN 네트워크 시스템을 기반으로 실험을 통하여 성능을 평가한다.

II. CAN 분석

1. 기본 CAN 프로토콜

CAN은 CSMA/CD 매체접근 기술을 기반으로 개발 되었으며, 네트워크 상에서 충돌을 해결하기 위해 송수신되는 메시지의 priority를 ID로 지정하는 기법을 사용하고 있다. CAN 프로토콜은 OSI 기준 모델을 기초로 하여 계층(layer) 형태로 구성된다. 그러나 OSI의 전 계층으로 구성되어 있지 않고 공장 자동화 환경과 관련된 부분만을 채택하여 다음의 3 계층으로 이루어져 있다.

- 1) 응용 계층(application layer): 사용자가 네트워크에 접근할 수 있도록 지원.
- 2) 데이터 연결 계층(data link layer): 물리적인 주소를 상하위 계층으로 연결.
- 3) 물리 계층(physical layer): 물리적인 매체를 통해 비트 흐름을 전송.

본 논문에서는 데이터 연결 계층을 활용하여 전송시간 지연 문제를 해결하고자 하며, 이 데이터 연결 계층을 이루는 MAC (Medium Access Control) 및 LLC (Logical Link Control) 중 LLC 부계층(sub-layer) 만을 활용한다.

2. 확장 CAN 프로토콜의 메시지 프레임

확장 CAN 프로토콜의 전체 메시지 프레임은 그림 2에서 보듯이 header 부분과 tail 부분으로 나누어진다. Header 부분은 메시지의 시작을 나타내는 SOF(Start of Frame)와 데이터 전송 중 충돌이 발생했을 때 충돌을 조정하는 arbiter field, 그리고 전송하고자 하는 데이터의 길이를 나타내는 control field로 구성된다. 그리고 Tail 부분은 실제 전송하고자 하는 데이터를 저장하는 data field, 전송된 데이터의 오류 감지를 위한 CRC Field 및 ACK Field, 그리고 프레임의 종료를 나타내는 EOF(End of Frame)로 구성된다.

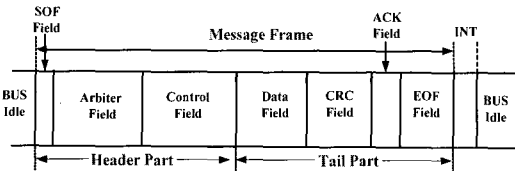


그림 2. 확장 CAN 프로토콜의 메시지 프레임.
Fig. 2. Message frame of the extended CAN protocol.

본 논문에서 전체 네트워크 시스템의 효율적인 대역 할당을 위해 제시되는 선행대기 열 기법에서 전체 메시지 프레임의 header 부분인 Arbitrator Field를 활용하여 priority를 재정의한다.

III. 동적 선행대기 열 기법(Dynamic precedence queue)

CAN은 네트워크 상에서 각각의 메시지의 priority를 나타내는 ID를 절대적으로 할당한다. 그러나, 이러한 기법은 몇 개의 unit이 동시에 전송을 시작할 때 일어나는 충돌을 해결할 수는 있으나 각각의 unit에 대해서 최대 허용 지연시간 범위를 보장할 수는 없다. 만약 네트워크 상에 매우 많은 unit이 연결되면 priority가 낮은 unit들은 전송 기회를 계속적으로 잃어버릴 수도 있다. 즉, priority가 높은 unit이 지속적으로 전송을 하면 priority가 낮은 unit은 상대적으로 급하지는 않지만 중요한 메시지를 오랫동안 전송하지 못하여 결국 위험한 상황을 야기할 수도 있다.

따라서, 전송되는 ID의 priority를 절대적으로 할당해 두면서도 부하의 상황에 따라 priority가 낮은 unit을 배려하는 상대적인 priority 값을 사용하는 기법이 필요할 것이다. 한 예로 서로 다른 unit들 사이에서 선행방법은 절대적인 priority를 가진 unit에게 통신 기회를 주되 낮은 순위의 unit도 일정 시간 내의 전송은 보장해줄 수 있는 것이다.

본 논문에서는 LLC의 프레임 수용 필터링 (frame acceptance filtering)을 변형하여 낮은 priority의 unit들의 전송을 보장해 줄 수 있는 방법을 제시한다. 이를 위하여, 메시지의 priority에 따라 지정되어 있는 ID를 수정 및 재정의 하여 하나의 대기 열에 넣어 줌으로써 결과적으로 네트워크 상의 메시지 전송을 원하는 unit들을 선행방식으로 서비스하게 한다. 이때 메시지 전송의 priority를 결정하기 위하여, 표준 CAN에서 사용하는 ID 및 DPQ 모드의 precedence를 함께 사용 한다. MAC은 전혀 변화가 없으며 기본적인 CAN 프로토콜과도 아무런 충돌 없이 호환이 가능하다.

1. 동적 선행대기 열 기법의 원리

본 논문에서 제시되는 CAN을 기반으로 하는 공정한 제어 기법의 기본 원리는 공유된 네트워크 상에서 전송하기를 원하는 모든 unit이나 비슷한 목적을 가진 각각의 unit을 하나의 대기 열에 입력시켜 순차적으로 전송되도록 하는 것이다. 그림 3에서 보듯이 전송이 지속적으로 지연되는 하나의 unit, H를 기준으로 그 unit에 함께 전송되어야 하는 unit들을 묶어서 대기 열을 생성한다. 따라서, 다수개의 대기 열이 분산적으로 생성될 수 있다.

그림 4는 최대 지연시간 초과로 인하여 대기 열을 생성하고자 하는 unit이 네트워크에 연결되어 있는 모든 unit들에게 대기 열 생성을 알리는 전송 프레임을 나타낸 것이다. 먼저

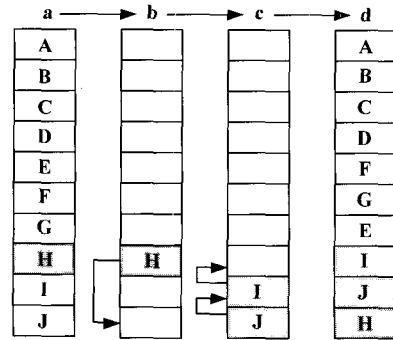


그림 3. DPQ 기법에서의 대기 열 생성.
Fig. 3. Generation of a precedence queue in DPQ mechanism.

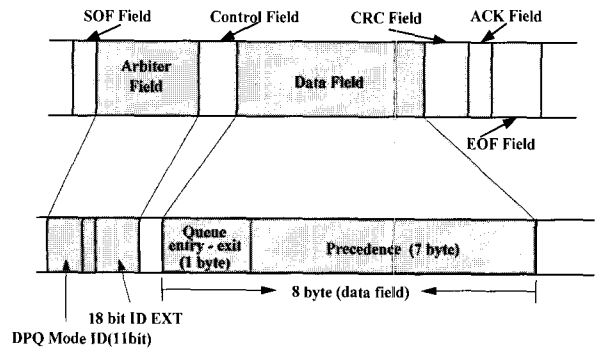


그림 4. DPQ 시작과 종료를 위한 메시지 프레임 구성.
Fig. 4. Structure of the message frame for DPQ Entry and exit.

대기 열을 생성하고자 하는 unit은 대기 열에 참여하는 unit들의 선행순위를 설정하여 선행순위가 높은 순으로 precedence field (7 byte)에 순차적으로 각각의 ID를 저장한다. 그리고, DPQ mode ID (11 bit) 및 queue entry (1 byte)를 구성하여 CAN 버스상에 전송한다. 각각의 unit은 DPQ mode ID 및 queue entry, 그리고 precedence value를 수신하여 자기가 대기 열에 참여하는 지 또는 precedence value이 얼마인지를 filtering하고 난 후 대기 열에 참여한다. 여기서 대기 열에 참여하는 unit들은 대기 열을 생성하는 unit들과 상호 연관성을 가지고 있어야 한다. 그리고, 위급한 상황이 종료되어 대기 열을 해제하고자 할 때 대기 열을 생성한 unit이 DPQ Mode ID(11 bit) 및 queue exit(1 byte)를 구성하여 CAN 버스상에 전송하고 난 후 대기 열에 참여한 각각의 unit들이 수신된 값을 filtering함으로써 DPQ mode가 종료되고 자기 자신의 priority로 되돌아간다. 따라서 대기 열을 생성하고자 하는 unit이 DPQ mode의 시작 및 종료에 관한 모든 권한 및 책임을 지고 있다.

그림 3에서와 같이 A에서 J까지의 unit으로 구성된 네트워크를 생각해 보자. 만약 unit H가 대기 열을 형성하고자 한다면 데이터 프레임에 대기 열에 들어갈 unit들의 ID를 그림 3의 하위 7 바이트 공간에 최대 7개까지 지정하여 전송해 줄 수 있다. 이 때 상위 바이트로부터 높은 선행순위(precedence value)를 가지게 된다. 그러면 각각의 unit들은 자기가 대기 열에 들어가는 지를 filtering하고 자기 자신의 선행순서를 할당받게 된다. 그리고 난 후 unit H는 실제 전송하기를 원하는 메시지를 전송하고 자기 자신은 대기 열의 마지막으로 가게

된다. 그리고, 나머지 unit들은 하나씩 대기 열에서 위로 올라간다. 그리고, 전송을 해야 하는 나머지 unit들을 모두 전송하고 대기 열을 해체할 것인지 아니면 유지할 것인지는 그림 4의 queue exit(1 byte)를 사용하여 unit H가 지정하여 준다.

이러한 선행방식을 사용하여 CSMA/CD에서 행하는 메시지 간의 충돌을 조정하는 역할을 수행하게 된다. 대기 열은 물리적으로 어떤 특정한 위치에 할당되지 아니하고, 네트워크 상에 전반적으로 분산되어 할당된다. 그리고 각각의 unit은 그들 자신의 대기 열을 저장하고 update를 해야 하는 의무를 가지고 있다. 즉, 자신의 최대 허용 지연시간이 다가오면 선행대기 열을 생성하여 관련된 unit들과 더불어 우선적으로 전송이 이루어질 수 있도록 동적으로 priority를 바꾸어 주어야 하고 위급한 상황이 끝나면 대기 열을 해체하여야 한다. 이러한 동적 선행대기 열 기법은 네트워크 상의 모든 unit들에게 대기 열을 생성할 수 있는 기회를 제공하고 있다. 그리고 다수 개의 대기 열이 존재하는 경우 대기 열 각각에 priority를 할당하여 각각 서로 독립적으로 동작할 수 있도록 한다.

여기서 priority와 선행순위와의 차이점에 주목해야 한다. 먼저 네트워크 상에 메시지를 전송할 때 priority는 각 전송 unit(메시지)에 할당되고, 반면 대기 열이 생성되는 경우 대기 열 내에서의 전송순위를 선행순위로 지정한다. 즉, priority는 메시지의 중요도 및 긴급성에 따라 고정적으로 할당하여 사용하는 것이며 선행순위는 대기 열 내에서의 전송순서를 나타낸다. 두 순위는 모두 충돌을 해결하기 위해서 사용되며 priority는 정적인 값인 반면에 선행순위는 사용자가 확인할 수 없으며 아주 동적인 값을 가진다.

메시지의 priority와 선행순위는 각각의 전송 프레임에서 18비트 ID 및 11 비트 ID의 위치에 할당 받게 되고 충돌이 발생했을 때 효과적인 해결을 할 수 있는 기법이다.

2. 동적 선행대기 열 기법의 실현 방법

동적인 선행대기 열 기법은 기본적인 CAN 프레임의 틀을 변화시키지 않고 실현할 수 있다. 선행대기 열을 지정하기 위해서 arbiter field를 사용한다. 그러나 표준 CAN의 arbiter field의 길이가 너무 짧기 때문에 확장 CAN의 arbiter field를 사용하여 동적 선행대기 열 기법을 실현한다.

그림 5는 확장 CAN 프레임의 header 부분을 나타내고 있다. 확장 CAN 프레임의 arbiter field는 총 31 비트로 구성되어 있고 상위 11 비트는 기본 ID를 나타내고 하위 18 비트는 확장 ID를 나타낸다. SSR(Substitute Remote Request)은 표준 CAN 프레임과 확장 CAN 프레임을 중재하는 역할을 담당하고 있으며 항상 "1"의 값을 전송한다. 만약 표준 CAN 프레임과 확장 CAN 프레임이 같은 기본 ID를 가지고 있으면 표준 CAN 프레임이 priority를 가진다. IDE(Identifier Extension)는 확장 CAN 프레임의 기본 ID와 확장 ID를 구분하기 위한 공간으로써 항상 "1"의 값을 전송하여 두 개의 ID field를 구분한다. RTR(Remote Transmission Request)는 원격 전송요구 비트로써 버스상에 어떤 한 unit이 다른 unit으로 데이터 전송을 요청할 때 "1"의 값을 전송함으로써 데이터 요청을 할 수 있다. r0, r1는 reserve 비트로서 향후에 사용하기 위한 예약된 비트들이다. 그리고 DLC(Data Length Code)는 실제 전송하고자

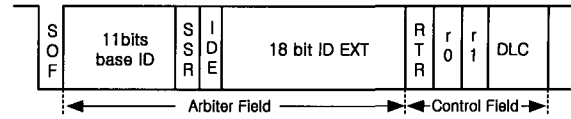


그림 5. 확장 CAN 프레임의 헤더 부분의 구성형태.
Fig. 5. Format of the header of extended CAN frames.

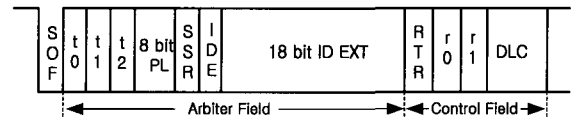


그림 6. DPQ 프레임 헤더 부분의 구성 형태.
Fig. 6. Header format of the DPQ frame.

하는 데이터의 길이를 나타낸다.

동적 선행대기 열 기법은 확장 CAN 프레임에서 arbiter field의 최상위 11 비트 기본 ID field를 사용하여 선행순위를 설정하며 상세한 비트 지정은 그림 6으로 보여진다. 표준 CAN에서 사용되는 ID는 18 비트의 EID(extended identification)로 옮겨지고 arbiter field의 최상위 11 비트는 동적 선행대기 열 기법에서 새롭게 동적으로 지정하는 선행순위 값을 저장하게 된다.

그림 6에서 처음 비트 t0는 논리적으로 항상 0의 값으로 설정하여 표준 CAN protocol과 동적 선행대기 열 기법에 의한 protocol을 구분한다. 따라서, 동적 선행 대기 열 기법은 표준 CAN protocol보다 항상 높은 priority를 가지게 되고 서로 공존할 수가 있다. t1, t2는 선행순위 대기 열을 사용할 때 높은 priority를 가지는 대기 열에 00의 값을 가장 낮은 priority를 가지는 대기 열에 11의 값으로 할당함으로써 대기 열들의 priority를 설정하는 공간이며 총 4개의 대기 열을 생성할 수 있다. 나머지 8 비트(8 bit PL)는 선행순위를 나타낸다. 즉, 8 비트가 전송대기 순서를 나타내는 것이다. 따라서, 최고 256개의 unit이 대기 열에 참여할 수 있다. 본 실험에서 사용된 동적 선행순위 대기 열 기법은 t0을 사용하여 standard CAN protocol과 구별하였으며 t1, t2를 사용하여 각각의 대기 열의 priority를 설정하며 8 비트를 사용하여 대기 열 내의 선행순위를 결정하였다.

IV. 시스템 구성 및 실험

본 논문에서 제시된 DPQ mechanism의 성능을 검증하기 위하여 실제 자동차의 네트워크 시스템에 사용되고 있는 unit을 기반으로 실험용 네트워크를 구성하였다. 먼저 powertrain 부분은 ENG ECU(Engine ECU), ETS(Electronic Throttle system), TCU(Transmission Control Unit)로 구성하고, chassis 부분은 ABS(Anti-Lock Brake System), LCU(Light Control Unit), 마지막으로 body 부분은 DDM(Driver Door Module), ADM(Assist Door Module), RAM(Rear Area Module), FSM(Front Sensor Module), RSM(Rear Sensor Module) 등 총 10개의 unit으로 전체 CAN 네트워크 시스템을 구성하였다. unit들의 priority는 차량의 주행에서 가장 중요한 engine과 관련이 있는 powertrain 부분, 즉 ENG ECU, ETS, TCU를 priority 1, 2, 3으로 각각 설정하여 전송하고자 하는 데이터를 최우선적으로 전송할 수 있도록

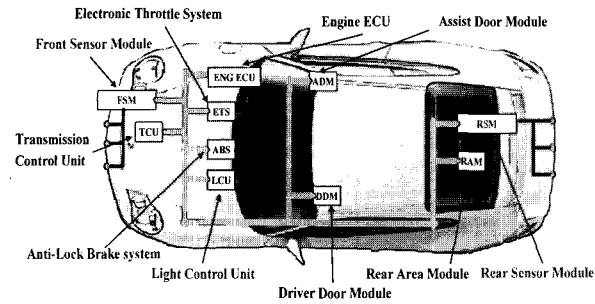


그림 7. 자동차 내부의 CAN 네트워크 시스템 구조.
Fig. 7. CAN network system architecture in vehicle.

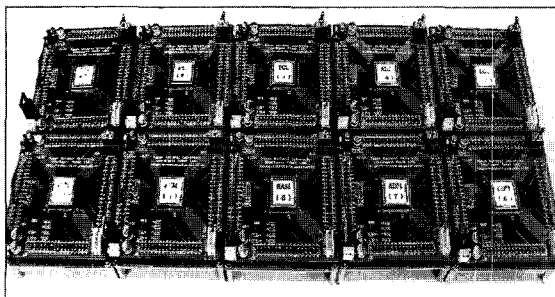


그림 8. 실험 시스템 구성도.
Fig. 8. Experimental system organization.

표 1. ID 및 우선순위 정의.

Table 1. Identification (ID) and priority definition.

Node (Priority)	Standard CAN ID	DPQ Mode CAN ID (1)	DPQ Mode CAN ID (2)
ENG ECU (1)	1 00 0000 0001	1 00 0000 0001	1 00 0000 0001
ETS (2)	1 00 0000 0010	1 00 0000 0010	1 00 0000 0010
TCU (3)	1 00 0000 0011	1 00 0000 0011	1 00 0000 0011
ABS (4)	1 00 0000 0100	1 00 0000 0100	0 10 0000 0001
LCU (5)	1 00 0000 0101	1 00 0000 0101	0 10 0000 0010
DDM (6)	1 00 0000 0110	1 00 0000 0110	1 00 0000 0110
ADM (7)	1 00 0000 0111	1 00 0000 0111	1 00 0000 0111
RAM (8)	1 00 0000 1000	1 00 0000 1000	1 00 0000 1000
FSM (9)	1 00 0000 1001	0 01 0000 0001	0 01 0000 0001
RSM (10)	1 00 0000 1010	0 01 0000 0010	0 01 0000 0010

DPQ Mode Entry & Exit definition	
DPQ Mode ID	11 1 0000 0000
DPQ Mode Entry	0000 1111
DPQ Mode Exit	1111 0000

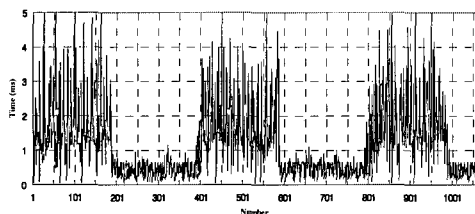


그림 9. DPQ mode와 표준 CAN mode의 스위칭에 따른 FSM (front sensor module)의 전송 지연시간.
Fig. 9. Transmission delay time of FSM to change the standard CAN mode and the DPQ mode.

하였고 chassis 부분인 ABS, LCU는 priority 4,5로 설정하며, 차량의 주위 장애물 유무 및 장애물과의 거리정보를 알려주는 FSM, RSM을 priority 9,10으로 각각 설정하였다. 그리고, 차량이 주차할 때 혹은 저속으로 운행 중일 때 상대적으로 낮은 priority로 인해 불규칙한 전송기회 및 전송 데이터의 지연시간이 발생하는 FSM, RSM unit에 DPQ mechanism을 적용하여 불규칙한 전송기회를 보장하고 제한된 시간 이내에 차량의 주위 장애물 정보를 안정적으로 획득할 수 있도록 하였다. 그리고, ABS, LCU에 추가적으로 DPQ Mechanism을 적용하여 전체 네트워크 시스템에 두 개의 대기 열을 생성하여 그 성능을 평가하였다.

각각의 unit은 CAN module이 내장된 TMS320LF2407을 사용하였으며 PCA82C251(Philips사)를 CAN transceiver로 사용하였다. unit의 전송속도는 250 Kbps로 설정하였으며 unit들의 데이터 전송 시 충돌이 자주 발생하도록 unit의 전송주기를 2 ms로 설정하였다.

표 1은 CAN 네트워크 시스템의 ID 및 priority를 정의하고 있다. 차량의 네트워크 시스템 중 power manager 부분인 engine ECU, ETS, TCU와 chassis 부분인 ABS, LCU 그리고 body manager 부분인 DDM, ADM RAM, 차량의 외부 장애물 유무 및 거리를 감지하는 sensor module로 구성된다. power manager unit은 차량의 운행에 있어 가장 중요한 engine과 연관된 부분을 제어하기 때문에 다른 unit에 비해 priority를 높게 설정하여 최우선적으로 데이터를 전송할 수 있도록 하였고, chassis unit은 중간 priority를 가지도록 설정하였고, body manager unit은 운전자 및 탑승자의 편의를 위해 제공되는 제어 unit으로서 전체시스템 중 낮은 priority를 가지도록 설정하였다. 특히 실험 1에서는 body manager unit 중 sensor 부분을 priority 9,10으로 설정하였으며 과부하 시 발생하는 지연시간을 보장하고 정해진 시간 이내에 FSM, RSM의 통합 데이터를 주기적으로 획득할 수 있도록 DPQ mode를 적용하였다. 그리고 실험 2에서는 실험 1에서 FSM, RSM로 구성된 대기 열 이외에 ABS, LCU unit으로 또 하나의 대기 열을 생성한다. 총 2개의 대기열을 생성하여 각각의 대기 열의 priority에 의한 전송 지연시간의 상대적인 변화를 실험을 통하여 평가하였다.

1. 실험 1

전체 10개의 unit으로 구성된 자동차 네트워크 시스템에서 차량이 서행 중이거나 주차를 할 경우에는 주기적이고 안정적인 외부 장애물의 위치 정보를 수집이 필요하다. 그러나, 데이터 전송량이 급격히 증가할 경우 상대적으로 낮은 priority를 갖는 FSM, RSM 등의 unit은 잦은 전송 충돌에 의해서 지연시간이 발생한다. 따라서, 실험 1은 주기적이며 안정적인 위치정보를 획득하기 위해서 FSM, RSM에 DPQ 기법을 적용하여 제안된 기법의 성능을 평가하였다.

그림 9는 DPQ mode와 표준 CAN mode의 스위칭에 따른 FSM의 전송 지연시간의 변화를 나타내고 있다. 구간 1~200, 401~600, 801~1000은 표준 CAN mode의 전송 지연시간을 나타내며, 201~400, 601~800 구간은 DPQ mode 적용 시 변화된 전송 지연시간을 나타낸다. 표준 CAN mode 시 평균 전송 지연시간이 2ms이고 FSM, RSM에 DPQ mode를 적용한 결과 priority 재정의에 의해서 0.4ms의 평균 전송 지연시간이 발생한다는 것을 알 수 있다.

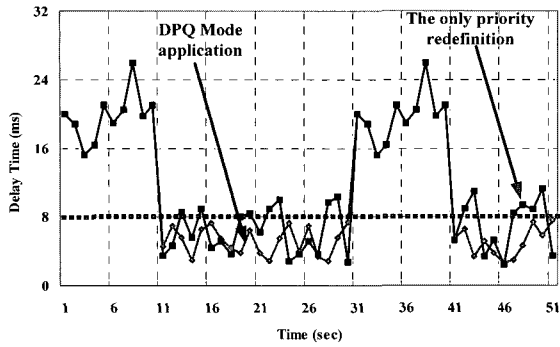


그림 10. 통합 데이터(FSM, RSM)의 획득시간.
Fig. 10. Integration data (FSM, RSM) obtaining time.

그림 10은 limit delay time을 8ms로 설정한 후 priority가 9.10인 FSM, RSM의 통합 데이터 정보획득 시간을 나타내고 있다. DPQ mode를 적용하지 않은 경우 이들 unit의 통합 데이터 정보획득 delay time은 평균 20ms로서 FSM, RSM의 데이터가 전송되지 못한다는 것을 알 수 있다. 하지만 본 논문에서 제시한 DPQ mode를 적용한 경우 FSM, RSM의 통합 데이터 획득시간이 4ms가 되는 것을 알 수 있다. 따라서 DPQ mode를 적용한 경우 선행방식에 의해서 차례로 데이터를 송신하기 때문에 8ms이내의 안정적인 주기적인 통합 데이터를 획득할 수 있다. 단지 priority를 재정의한 경우에는 random한 데이터 송수신이 이루어지기 때문에 제한된 시간 이내에 FSM, RSM의 통합 데이터 획득을 보장할 수 없음을 주목해야 한다.

따라서, 실험 1에서는 차량의 저속 운행 및 주차 시 상대적으로 낮은 priority를 갖는 FSM, RSM에 DPQ mode를 적용함으로써 주기적이고 안정적인 차량의 주위 장애물 정보 획득을 할 수 있다는 것을 확인하였다.

2. 실험 2

실험 1에서는 하나의 대기 열을 이용하여 상대적으로 낮은 priority를 가지는 unit들의 전송 지연을 해결하였다. 하지만 DPQ mode가 실행되고 있는 동안 재정의된 priority에 의해 하위 priority를 가지는 unit들의 전송 지연이 발생하게 된다. 즉, 차량의 FSM, RSM unit에 DPQ mode를 적용한 경우 ABS, LCU와 같은 중간 priority를 가지는 unit의 priority가 낮아지게 된다. 차량의 야간 운행 또는 급제동 시 이들 unit의 데이터가 필요하다. 따라서 실험 2에서는 차량의 야간 운전 시 운전자의 넓은 시야 확보 및 안정적인 제동을 위해서 자동차의 chassis 부분인 ABS, LCU에 대해 DPQ mode를 적용하여 보다 빠르고 안정적으로 ABS, LCU를 제어할 수 있도록 실험을 하였다. 전체적으로 실험 1에서 FSM, RSM로 구성된 대기 열 이외에 ABS, LCU unit으로 또 하나의 대기 열을 생성하여 실험을 수행한다. 전체 네트워크 상에 총 2개의 대기 열이 서로 독립적으로 수행이 되며 2개의 대기 열들은 상대적인 priority에 의해서 각각의 대기 열들의 전송 지연시간의 변화를 실험한다.

그림 11은 두 개의 대기 열 생성에 따른 unit들의 전송 지연시간의 변화를 나타낸 것이다. 1~10는 표준 CAN mode를 사용한 시간을 나타내고, 11~30은 FSM, RSM unit으로 대기 열을 생성한 구간이며, 31~60은 ABS, LCU unit으로 다른 하나의

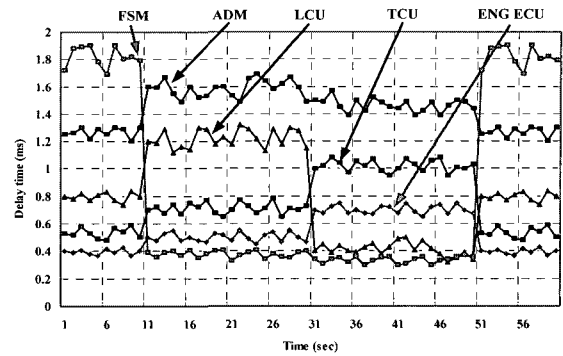


그림 11. 두 개의 대기 열 생성에 따른 unit들의 전송 지연시간.
Fig. 11. Transmission delay time of units to generation two precedence queue.

대기 열을 생성하여 전체 네트워크상에 두 개의 대기 열이 존재하는 구간이다. 대기 열을 생성한 후 선행적으로 데이터를 전송함으로써 상대적으로 전송 지연시간이 빨라졌고, 그 외의 unit은 상대적으로 전송 지연시간이 늘어났음을 확인할 수 있었다. 실험 2에서는 FSM, RSM unit으로 생성된 대기 열과 ABS, LCU unit으로 생성된 대기 열을 생성하여 네트워크의 동작상태를 확인하였다. 즉, 차량의 주차 및 서행 운행 시 낮은 priority를 가지는 차량의 body 부분(FSM, RSM)과 차량의 야간 운행, 약천후 시 중간 priority를 가지는 차량의 chassis 부분(ABS, LCU)에 DPQ mode를 적용하여 실 시간적인 정보 획득으로 인해 차량의 안정적인 운행 및 돌발상황에 대한 대처가 가능하다는 것을 확인하였다.

실험 2에서는 FSM, RSM에 의해 생성된 대기 열이 ABS, LCU에 의해서 생성된 대기 열보다 상대적으로 높은 priority를 갖고 있기 때문에 두 개의 대기 열이 동시에 발생된 31~50 구간에서 FSM의 전송 지연시간 (0.38ms)이 ABS의 전송 지연시간(0.41ms)보다 빠르고 각각 독립적으로 수행된다는 것을 실험을 통해 확인할 수 있었다.

V. 결론

본 논문은 표준 CAN 프로토콜을 사용하는 자동차 네트워크 시스템 상에서 일어나는 충돌을 조정하고 고정된 priority 기법에 의해 발생하는 비효율성을 개선 하기 위하여 DPQ 기법을 제안하였다. 전체 시스템은 실제 차량에서 사용되고 있는 ENG ECU, ETS, TCU (powertrain part), ABS, LCU(chassis part), 그리고, DDM, ADM, RAM, FSM, RSM(body part) 등 총 10개의 unit으로 전체 네트워크 시스템을 구축하여 그 유용성을 검증하였다. 특히, 차량의 서행 운전 및 주차 시 낮은 priority에 의해서 비 주기적이고 전송 지연시간 초과로 인한 데이터 손실을 보정하기 위하여 FSM, RSM에 DPQ 기법을 적용하여 차량 주위환경 정보를 주기적 이며 안정적으로 전송하는 것을 실험 1을 통하여 확인 하였다.

그리고, 차량의 야간 운행 시 운전자의 넓은 시야 확보 및 안정적인 제동을 위하여 차량의 chassis part인 ABS, LCU에 DPQ 기법을 적용하여 그 성능을 평가 하였다. 특히, 실험 2에서는 상대적으로 낮은 priority를 갖는 chassis part, body part로 이루어진 unit들 사이에 두 개의 대기 열을 생성하여 DPQ

기법을 적용함으로써 차량의 야간 운행 및 주차와 같은 상황에서 주기적인 차량 주위환경 정보 획득에 의해서 운전자가 안정적으로 차량을 제어할 수 있도록 하였다.

그러나 priority가 높은 unit에 DPQ 기법을 적용했을 경우 표준 CAN 프로토콜을 적용했을 때의 전송시간 보다 개선되는 효과는 보기가 어려웠다. 또한 priority가 낮은 unit에 DPQ 기법을 적용한 경우 이들 unit의 데이터 전송이 종료된 후 DPQ 기법이 종료되므로 priority가 높은 unit의 전송 지연 시간 증가로 인한 시스템 성능 저하와 같은 문제는 발견할 수 없었다. 향후 과제로는 동적으로 DPQ 기법을 적용하여 각 unit의 지연시간을 효율적으로 관리하는 알고리즘을 개발할 것이며 이 개발된 알고리즘을 다른 CAN 응용 시스템과 호환에 있어서 얼마나 편리하게 적용할 수 있는가를 연구할 것이다.

참고문헌

- [1] T. Inoue et al., "Protocol for automotive local area network (PALMINET)-a newly developed in-vehicle communication system based on SAEJ1850," *SAE Paper 890535*, 1989.
- [2] H. Kimura and Y. Himono et al., "The development of the advances protocol for automotive local area multiplexing network (Advanced PALMINET)," *SAE Paper 940365*, 1994.
- [3] International Standard Organization, "Road-vehicles Interchange of digital information - Controller area network for high-speed communication" *ISO 11898*, November 1993.
- [4] International Standard Organization, "Road-vehicles Interchange of digital information - Controller area network for high-speed

- communication" Draft Amendment, *ISO 11898:1993/DAM 1*, February 1994.
- [5] CAN in AUTOMATION International Users and Manufacturers Group e. V. "CAN Application Layer (CAL)," CiA/DS201-CiA/DS205, CiA/DS207.
- [6] 박진우, 노동규, 박재한, 허화라, 이장명, "CAN을 이용한 분산 제어 구조를 가지는 이동 로봇 구현" 추계 합동 학술논문 발표회 논문집. pp.251-255. 1999.
- [7] 홍성수, "분산 실시간 제어 시스템 개발," CASE 기술 특집: 실시간 제어 시스템(3), *ICASE*, 1, 1998.
- [8] IEEE Standards for Local Area Networks, "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications," *ANSI/IEEE Std 802.3-ISO/DIS 8802/3*, 1985.
- [9] K. Tindell, A. Burns, and A. Wellings, "Calculating controller area network (CAN) message response times," in *Proc. 1994 IFAC Workshop on Distributed Computer Control Systems*, Toledo, Spain, September 1994.
- [10] K. Tindell and A. Burns, "Guaranteeing message latencies on control area network (CAN)," in *Proc. 1st International CAN Conference*, Mainz, Germany, September 1994.
- [11] K. Tindell, A. Burns, and A. Wellings, "Analysis of hard real-time communications," *Report YCS 222, Department of Computer Science*, University of York, to appear in *Real-Time Systems*, 1994.
- [12] "SDS-Smart distributed system specification" *Hineysell Inc.*, Micro Switch Division, Phoenix, AZ, GS 052-103/104/1-5/106/107/108.



최 호 식

2001년 부경대학교 제어계측공학과 졸업. 2003년~현재 부산대학교 대학원 전자공학과 석사과정 재학중. 관심분야는 로봇제어 최적 알고리즘개발, CAN Network, Teleoperation.



이 장 명

1980년 서울대학교 전자공학과 졸업. 1982년 동 대학원 졸업. 1990년 USC(미국 남가주 대학교) 공학박사. 1992년~현재 부산대학교 정교수. 관심분야는 지능로봇 시스템의 설계 및 제어, 마이크로 프로세서 응용시스템 설계, 센서 및

모터 구동 제어.



최 효 식

2005년 부산대학교 전자전기통신 공학부 졸업. 2005년~현재 부산 대학교 대학원 전자공학과 석사과정 재학중. 관심분야는 지능제어시스템, CAN Network, 휴머노이드 로봇.