

## CAN 네트워크상의 효율적인 대역 할당을 위한 분산 선행대기 열 기법

### A Distributed Precedence Queue Mechanism to Assign Efficient Bandwidth in CAN Networks

최 호 식, 이 장 명\*

(Ho-Seek Choi and Jang-Myung Lee)

**Abstract :** This paper presents a distributed precedence queue mechanism to resolve unexpected transmission delay of a lower priority transaction in a CAN based system, which keeps a fixed priority in data transactions. The mechanism is implemented in the upper sub-layer of the data link layer (DLL), which is fully compatible with the original medium access control layer protocol of CAN. Thus the mechanism can be implemented dynamically while the data transactions are going on without any hardware modification. The CAN protocol was originally developed to be used in the automotive industry, and it was recently applied for a broader class of automated factories. Even though CAN is able to satisfy most of real-time requirements found in automated environments, it is not to enforce either a fair subdivision of the network bandwidth among the stations or a satisfactory distribution of the access delays in message transmissions. The proposed solution provides a superset of the CAN logical link layer control, which can coexist with the older CAN applications. Through the real experiments, effectiveness of the proposed mechanism is verified.

**Keywords :** controller area network (CAN), a fair subdivision of the network bandwidth, data link layer

#### I. 서론

CAN 프로토콜은 차량 내부의 복잡한 배선 문제 및 이로 인한 신뢰도 저하를 해결하기 위해서 개발되었다[1,2]. 이 프로토콜의 유용성으로 인하여 다양한 산업분야에 적용되어 신뢰성이 높은 네트워크를 구축하고 있다[3,4,5].

다양한 산업 현장에서 사용되고 있는 CAN 기법의 가장 큰 이점은 Intel사의 82527 CAN 제어기와 같은 전자 부품들의 가격이 저렴하다는 것이다. 사용하기 간편하고 가격이 저렴하다는 것은 자동화 네트워크를 구축함에 있어서 가장 우선적으로 고려되어야 할 사항들이기 때문에 CAN의 보급이 지속되고 있다. 또한, CAN은 메시지의 전송에 우선 순위를 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.

CSMA/CD 프로토콜[6]을 기반으로 하는 IEEE 802.3 표준 access 기술과는 다르게, CAN에서 사용하는 CSMA/CD 매체 접근 기술은 충돌이 발생했을 때, 신호들을 파괴시키지 않고 충돌을 조정하는 조정자가 가장 우선순위가 높은 프레임은 전송하고, 나머지는 전송을 중지시킨다. 전송되는 프레임은 특별한 목적지를 가지고 있지 않으며 전체 네트워크 상에서 메시지에 따라 유일한 ID를 부여 받는 목적코드를 가진다. 이러한 ID는 네트워크상에 전체적으로 알려져 있어야 하며

각각의 개체는 관심대상이 되는 값들을 선별적으로 수신해서 사용할 수 있다. CAN은 이러한 ID를 사용해서 네트워크 상에서 전송되는 다른 종류의 메시지 혹은 개체에 대해 우선 순위를 절대적으로 할당한다. 이러한 기법은 네트워크 상에서 충돌이 일어났을 때 충돌을 조정하는 좋은 방법이 된다.

그러나 통신도중 충돌이 발생하면 CAN 기법의 사용으로 인해서 우선순위가 높은 개체는 송신을 계속 수행하지만 우선순위가 낮은 개체는 송신을 하지 못하고 대기해야 한다. 만약 CAN 네트워크 상에 과부하가 걸리게 되면 우선순위가 낮은 개체는 우선순위가 높은 개체 때문에 계속적으로 송신이 저지되어 지연시간[7,8,9]이 길어지게 된다. 이러한 동작은 round-robin과 같은 응용프로그램을 사용하고 정기적인 통신을 행하는 네트워크 상에서는 수용될 수도 있으나 각각 독립적이고 비정기적으로 통신을 하는 네트워크상에서는 큰 문제점이 될 수 있다.

실제 네트워크 상에 과부하가 걸리면 데이터 전송 충돌의 수가 증가하여 실제 데이터 전송량은 급격히 감소한다. 이러한 상황이 계속되면 네트워크는 매우 불안정한 상태가 되고 장시간 동안 통신 불능 상태를 야기할 수도 있다. 과부하 시 CAN 네트워크 상에 정적으로 할당되는 ID로 인하여 데이터 전송의 비효율성을 제거하고 최대 허용 지연시간을 보장하기 위해서 본 논문에서는 네트워크 상에 과부하가 발생할 때 우선순위가 상대적으로 낮고 전송되는 데이터의 목적이 비슷한 개체들에 대해 각각 선행대기 열을 할당하여 네트워크 상의 분산된 대기 열 들에게 공정한 전송기회와 지연시간을 최대한으로 줄일 수 있는 기법을 제시하였다[10].

이러한 선행대기 열들은 정적으로 할당하는 것이 아니라 시스템의 전송부하 조건에 따라 동적으로 할당되어 네트워크 상의 전송 효율을 높일 수 있다. 그리고 각각의 대기 열

\* 책임저자(Corresponding Author)

논문접수 : 2004. 9. 25., 채택확정 : 2004. 10. 13.

최호식, 이장명 : 부산대학교 전자공학과

(chs\_chs@hanmail.net/jmlee@pusan.ac.kr)

※ 본 연구는 정보통신부 대학 IT 연구센터 육성·지원사업의 연구결과로 수행되었습니다.

들은 상대적인 우선순위를 가지고 있으며 서로 독립적으로 데이터 전송순서를 할당 할 수 있다.

따라서 본 논문에서는 네트워크의 부하가 극심하여 우선 순위가 낮은 개체의 전송이 지속적으로 지연되는 경우에 이 개체의 우선순위를 동적으로 조정하여 일정시간 이내에 전송이 가능하도록 하는 기법을 확장 CAN 프로토콜을 기반으로 개발하고자 한다.

하나 또는 여러 개의 개체들이 공유된 네트워크를 독점하는 것을 방지하고, 허용 지연시간을 넘기지 않도록 모든 개체들을 관리하기 위한 방법으로 제시하는 것으로, 3계층으로 나누어진 CAN의 중간 계층인 데이터 연결 계층 (data link layer) 에서 실현된다.

CAN 프로토콜에서는 ID가 정적으로 할당되고, 이 고정된 ID에 의해서 충돌 문제를 해결하기 때문에 이러한 두 개의 요구조건(공정한 전송기회, 지연시간)을 만족하게 할 수 없다. 본 논문에서는 각각의 전송개체의 ID에 따른 입력 프레임 필터링 하고 CAN의 Identifier field에서 ID를 재정의 함으로서 주어진 문제를 해결할 수 있음을 보여준다. 이를 위하여 분산된 선행 대기 열을 확장 CAN의 ID field를 사용하여 지정해 줌으로써 각각의 개체들을 공정한 전송순서에 따라 전송하고 최대 허용 지연시간을 만족하도록 한다.

이러한 변환은 CAN의 MAC 부-계층 (sub-layer)에 의해서 제공되는 부분과 서로 호환이 가능하기 때문에 표준 CAN 프로토콜에서 발전된 상업적인 응용 분야에도 동일하게 적용할 수 있다. 즉, 본 논문에서 제시된 기법은 CAN 기반의 모든 응용기법과 호환적으로 사용될 수 있다.

## II. CAN 분석

### 1. 기본 CAN 프로토콜

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

- 응용 계층 (Application Layer)  
; 사용자가 네트워크에 접근할 수 있도록 지원.
- 데이터 연결 계층 (Data Link Layer)  
; 물리적인 주소를 상-하위 계층으로 연결.
- 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 부-계층 (sub-layer)만을 활용한다.

### 2. CAN 의 물리 계층

CAN 네트워크는 버스 토폴로지의 형태로 구성된다. 그리고 버스는 두 개의 상호 보완적인 Dominant와 Recessive의 값으로 구성된다.

Dominant 값은 논리적으로 0의 값을 갖고 Recessive 값은 1

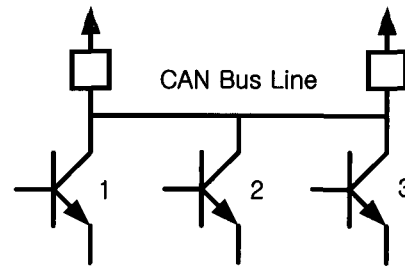


그림 1. CAN의 물리 계층.

Fig. 1. CAN physical layer.

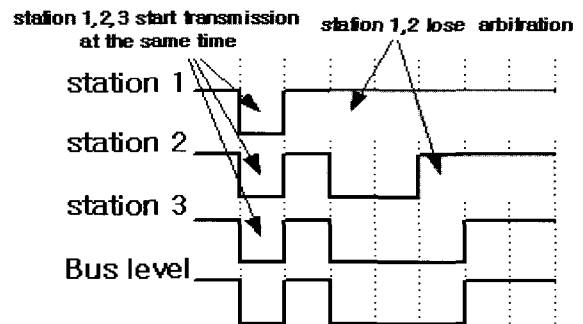


그림 2. CAN 조정자 위상.

Fig. 2. CAN arbitration phase.

에 상응하는 값을 갖는다. 동시에 몇 개의 개체가 Dominant와 Recessive 값을 전송 한다면 버스상의 결과 값은 0의 값인 Dominant 값 만이 전송된다. 다시 말하면 CAN은 전자회로에서 open-collector와 유사한 방법으로 구성된다.

### 3. LLC 부-계층

LLC 부-계층은 데이터 연결 계층의 상위 부분으로 프레임 필터링 및 recovery 기능을 담당한다. LLC 부-계층 하부의 MAC 부-계층은 수신된 프레임의 Identifier field에서 어떤 값을 체크하지는 않고 간단히 수신된 값만을 읽는다. CAN 프로토콜에서 ID는 메시지의 목적지 주소를 나타내는 것이 아니고 메시지의 종류를 나타낸다는 것을 주의해야 할 것이다.

LLC 부-계층은 수신된 프레임을 수용할 지 아니면 수용하지 않을 지를 결정하는 기능을 가지고 있다. 또한, LLC는 충돌이 발생하여 전송자격을 잃었는지 아니면 에러가 발생해서 프레임을 재전송 해야 할 지를 조정하는 역할을 담당한다. 그리고 사용자는 메시지가 성공적으로 전송될 때까지 전송 결과를 알지 못한다.

## III. 분산된 선행 대기 열 기법 (DPQ)

CAN은 네트워크 상에서 각각의 메시지의 우선순위를 ID로 절대적으로 할당한다. 그러나, 이러한 기법은 몇 개의 개체가 동시에 전송을 시작할 때 일어나는 충돌을 해결할 수는 있으나 각각의 개체에 대해서 최대 허용 지연시간 범위를 보장할 수는 없다. 만약 네트워크 상에 매우 많은 개체가 연결되면 우선순위가 낮은 개체들은 전송 기회를 계속적으로 잃어버릴 수도 있다. 즉, 우선 순위가 높은 개체가 지속적으로 전송을 하면 우선순위가 낮은 개체는 상대적으로 급하지는

않지만 중요한 메시지를 오랫동안 전송하지 못하여 결국 위험한 상황을 야기할 수도 있다.

따라서, 전송되는 ID의 우선순위를 절대적으로 할당해 두면서도 부하의 상황에 따라 우선순위가 낮은 개체를 배려하는 상대적인 우선순위 값을 사용하는 기법이 필요할 것이다. 한 예로 서로 다른 개체들 사이에서 round-robin 방법은 절대적인 우선 순위를 가진 개체 순서를 통신의 기회를 주되 낮은 순위의 개체도 일정 시간 내의 전송은 보장해줄 수 있는 것이다.

본 논문에서는 LLC의 프레임 수용 필터 링 (frame acceptance filtering)을 변형하여 낮은 우선 순위의 개체들의 전송을 보장해 줄 수 있는 방법을 제시한다. 이를 위하여, 메시지의 우선 순위에 따라 지정되어 있는 ID를 수정하여 재정의 하여 하나의 대기 열에 넣어 줌으로써 결과적으로 네트워크 상의 메시지 전송을 원하는 개체들을 round-robin 방식으로 서비스하게 한다. 이 때 메시지 전송의 우선순위를 결정하기 위하여, 표준 CAN에서 사용하는 ID 및 DPQ 모드 ID를 함께 사용한다. MAC은 전혀 변화가 없으며 기본적인 CAN 프로토콜과도 아무런 충돌 없이 호환이 가능하다.

1. DPQ(Distributed Precedence Queue) 원리

본 논문에서 제시되는 CAN을 기반으로 하는 공정한 제어 기법의 기본 원리는 공유된 네트워크 상에서 전송하기를 원하는 모든 개체나 비슷한 목적을 가진 각각의 개체를 하나의 대기 열에 입력시켜 순차적으로 전송되도록 하는 것이다. 그림 3에서 보듯이 전송이 지속적으로 지연되는 하나의 개체, C를 기준으로 그 개체에 함께 전송되어야 하는 개체들을 묶어서 대기 열을 생성한다. 따라서, 다수개의 대기 열이 분산적으로 생성될 수 있다. 본 연구에서는 두 개의 대기 열을 사용하는 것으로 한다.

이러한 분산대기 열 프로토콜은 네트워크상의 모든 개체들에게 대기 열을 생성할 수 있는 기회를 제공하고 있다. 그리고 다수 개의 대기 열이 존재하는 경우에는 대기 열 각각에 우선순위를 할당하여 각각 서로 독립적으로 동작할 수 있도록 한다.

그림 5의 11 비트 표준 ID 부분에 저장된 DPQ 모드 ID는 각각의 개체의 현재 자기자신의 대기 열 순서를 가리키게 된다.

하나의 개체가 전송을 완료할 때 마다 자기자신은 대기 열

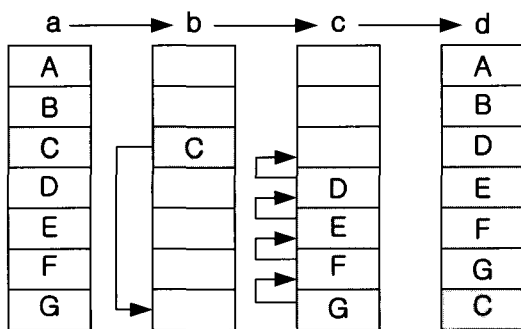


그림 3. DPQ 기법에서의 대기 열 생성.  
Fig. 3. Generation of a precedence queue in DPQ mechanism.

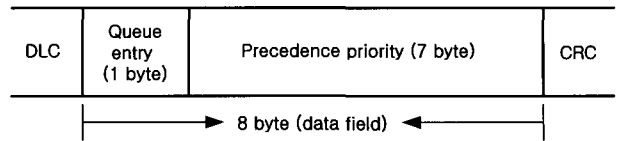


그림 4. DPQ를 위한 데이터 영역의 구성.  
Fig. 4. Structure of a data field for DPQ.

의 마지막으로 들어가고 다른 개체들은 하나씩 위로 이동하여 대기 열의 맨 위에 도달하면 전송된다. 이러한 round-robin 방식을 사용하여 CSMA/CD에서 행하는 메시지간의 충돌을 조종하는 역할을 수행하게 된다.

이러한 대기 열은 물리적으로 어떤 특정한 위치에 할당되지 아니하고, 네트워크 상에 전반적으로 분산되어서 할당된다. 그리고 각각의 개체는 그들 자신의 대기 열을 저장하고 update를 해야 하는 의무를 가지고 있다.

즉, 자신의 허용 최대 지연시간이 다가오면 선행 대기 열을 생성하여 관련된 개체들과 더불어 우선적 전송이 이루어질 수 있도록 동적으로 우선 순위를 바꾸어 주어야 하고 위급한 상황이 끝나면 대기 열을 해제하여야 한다.

그림 3에서와 같이 A에서 G까지의 개체로 구성된 네트워크를 생각해 보자. 만약 개체 C가 대기 열을 형성시키고자 하면 데이터 프레임에 대기 열에 들어갈 개체들의 ID를 그림 4의 하위 7 바이트에 의하여 7 개까지 지정하여 전송해 줄 수 있다. 이 때 상위 바이트부터 높은 선행순위(Precedence priority)를 가지게 된다. 그러면 각각의 개체들은 자기가 대기 열에 들어가는 지를 filtering하고 자기자신의 대기 열을 할당 받게 되고 개체 C는 실제 전송하기를 원하는 메시지를 전송하고 난 후 자기자신의 대기 열의 마지막으로 가게 된다. 그리고, 나머지 개체들은 하나씩 대기 열에서 위로 올라간다. 그리고, 전송을 해야 하는 나머지 개체들을 모두 전송 하고 난 후 대기 열을 해제할 것 인지 유지할 것인지는 그림 4의 상위 1 바이트를 사용하여 지정해 준다.

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

메시지의 우선순위와 선행순위는 각각의 전송 프레임에서 18비트 ID 및 11비트 ID의 위치에 할당 받게 되고 (그림 5 및 6 참조) 충돌이 발생했을 때 효과적인 해결을 할 수 있는 기법이다.

2. DPQ 실현 방법

분산된 선행대기 열 기법은 기본적인 CAN 프레임의 틀을 변화시키지 않고 실현할 수 있다. 선행대기 열을 지정하기 위해서 Identifier field를 사용한다. 그러나 표준 CAN의 Identifier field의 길이가 너무 짧기 때문에 확장 CAN의 Identifier field를 사용하여 분산된 선행순위 대기 열 기법을

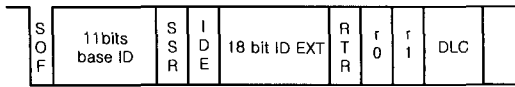


그림 5. CAN 확장 프레임의 헤더 부분의 구성형태.  
Fig. 5. Format of the header of extended CAN frames.

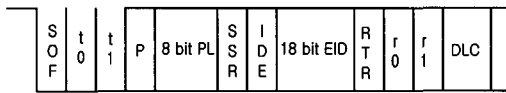


그림 6. DPQ 프레임의 헤더 부분 구성형태.  
Fig. 6. Format of the header of DPQ frames.

실현한다.

분산된 선행대기 열 기법은 11 비트(bit)의 Identifier field를 사용하여 선행순위를 설정하며 상세한 비트지정은 그림 6으로 보여진다. 즉, 표준 CAN에서 사용되는 ID는 18 비트의 EID로 옮겨지고 11 비트 Identifier field는 분산된 선행순위 대기 열 기법에서 새롭게 동적으로 지정하는 선행순위 값을 저장하게 된다.

그림 6에서 처음의 두 개의 비트 (t0, t1)는 논리적으로 0의 값으로 설정하여 표준 CAN 통신과 분산 선행대기 열 기법에 의한 프로토콜을 구분한다. 따라서, 분산 선행대기 열 기법은 기본적인 CAN 기법보다 항상 높은 우선순위를 가지게 되고 서로 공존할 수가 있다. P는 두 개의 선행순위 대기 열을 사용할 때 높은 우선순위를 가지는 대기 열에 0의 값을 낮은 우선순위를 가지는 대기 열에 1의 값으로 할당한다. 실제적으로 t1과 P를 활용하면 최대 대기 열을 4개까지 확장하여 우선 순위를 할당할 수 있다. 나머지 8 비트는 선행순위를 나타낸다. 즉, 이 8 비트가 전송대기 순서를 나타내는 것이다. 본 실험에서 사용한 분산된 선행순위 대기 열 기법은 t0, t1을 사용해서 표준 CAN 기법과 구별하였으며 P를 사용해서 각각의 대기 열의 우선순위를 설정하며 8비트를 사용하여 대기 열 내의 선행순위를 결정하였다.

IV. 시스템 구성 및 실험

본 논문에서 제시된 기법의 효용성을 검증하기 위하여 차량의 스로틀-바디(Throttle-body) 제어기에 사용되는 액츄에이터 (Actuator) ECU 및 실제 차량의 고장을 진단하고 각각의 센서 한계 값들을 설정할 수 있는 휴대용 검사기 ECU를 기본적인 개체로 설정하였고, 흡입연료 ECU, 점화 ECU, 사이드-미러 ECU, 배기구 ECU 등과 같은 차량 곳곳에 사용되는 ECU 등을 가상적인 개체로 추가하여 총 10개의 개체로 시스템을 구성하였다.

각각의 개체는 CAN 모듈이 내장된 TMS320LF2407를 사용하였으며 PCA82C251 (Philips 제조)를 CAN transceiver로 사용하였다. 각각의 개체의 전송속도의 250 Kbps로 설정하였다.

첫 번째 실험에서 전체 10 개의 개체에 대한 전송 주기를 10 ms 및 2 ms의 두 가지의 경우로 설정하였다. 즉, 전송 주기가 10 ms일 때는 버스 상에서 메시지 충돌이 자주 발생하지 않는 경우이며 2 ms일 때는 충돌이 자주 발생하는 경우가

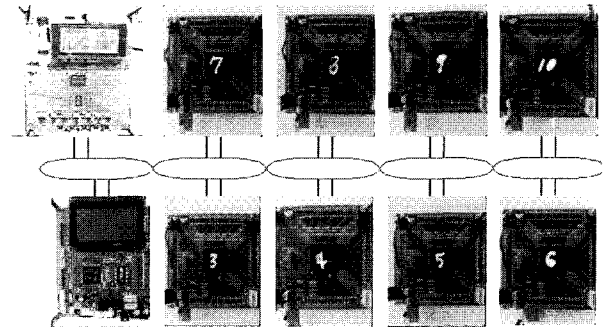


그림 7. 전체 시스템 구성도.  
Fig. 7. Total system organization.

표 1. 실험 1에 대한 ID 정의.

Table 1. Identification definition (ID) in experience 1.

node	standard CAN	DPQ
PORTABLE	11 1 0000 0001	11 1 0000 0001
MAIN ECU	11 1 0000 0010	11 1 0000 0010
3	11 1 0000 0011	11 1 0000 0011
4	11 1 0000 0100	11 1 0000 0100
5	11 1 0000 0101	11 1 0000 0101
6	11 1 0000 0110	11 1 0000 0110
7	11 1 0000 0111	11 1 0000 0111
8	11 1 0000 1000	00 1 1111 1101
9	11 1 0000 1001	00 1 1111 1110
10	11 1 0000 1010	00 1 1111 1111

표 2. 실험 2에 대한 ID 정의.

Table 2. Identification definition (ID) in experience 2.

node	standard CAN	DPQ
PORTABLE	11 1 0000 0001	11 1 0000 0001
MAIN ECU	11 1 0000 0010	11 1 0000 0010
3	11 1 0000 0011	11 1 0000 0011
4	11 1 0000 0100	11 1 0000 0100
5	11 1 0000 0101	11 1 0000 0101
6	11 1 0000 0110	11 1 0000 0110
7	11 1 0000 0111	11 1 0000 0111
8		00 1 1111 1101
9		00 1 1111 1110
10		00 1 1111 1111

된다. 각각의 전송 주기마다 개체 1 (Portable ECU), 개체 2 (Main ECU) 순으로 전송 메시지의 priority를 높게 설정하였으며 점차적으로 priority가 낮아지게 설정하였다. 전송 주기가 2 ms일 때는 버스 상의 메시지 충돌로 priority가 낮은 개체 8, 9 및 10은 전송 지연이 크게 일어나므로 이를 해결하기 위하여 DPQ Mode를 개체 8, 9 및 10에 적용하였다.

두 번째 실험은 전송주기가 2 ms 인 개체를 7 개를 사용하는 경우와 10 개의 개체를 사용하는 경우로 구분하여 사용하

여 시스템을 구성하였다. 실험 1에서와 같은 방법으로 개체 1 (Portable ECU), 개체 2 (Main ECU) 순으로 전송 메시지의 priority를 높게 설정하였으며 점차적으로 낮아지게 priority를 설정하였다.

개체가 7 개인 경우보다 개체가 10 개인 경우에 충돌이 자주 발생하도록 개체 수를 증가시켰다. 그리고 전송주기가 2 ms이고 개체가 10 개일 때 실험 1과 마찬가지로 개체 8, 9, 10에 DPQ Mode를 적용하였다.

**V. 결과 및 분석**

그림 8은 개체 1의 전송 지연 시간을 나타낸 것으로, X축의 숫자 1에서 50까지는 전송주기가 10 ms일 때의 전송 지연 시간을 나타낸 것이고 51에서 100까지는 전송주기가 2ms 일 때이고 101에서 150까지는 DPQ Mode를 적용했을 때의 값을 나타낸다.

그림 8의 결과에서 알 수 있듯이 개체 1은 전송 주기가 10 ms 일 때 보다는 2 ms일 때 잦은 전송을 위한 스위칭으로 인하여 지연 시간이 길어지는 것을 확인할 수 있고 DPQ Mode 일 때는 개체 8, 9 및 10에 대한 배려로 추가적인 지연 시간이 발생하는 것을 알 수 있다.

그림 9에서 보듯이 개체 8의 경우 전송 주기가 2 ms 인 경우에 다른 개체들에 비해 상대적으로 priority가 낮기 때문에 긴 지연시간이 발생하는 것을 알 수 있다. 이를 극복하기 위하여 priority를 영구적으로 바꾸는 대신에 동적으로 DPQ Mode를 적용 했을 때 그림 10에서 보듯이 다른 개체들보다 지연 시간이 월등히 향상되고 전송기회가 보장 되는 것을 확인할 수가 있었다.

그림 11의 실험 2의 경우의 전송 지연 시간을 보여주는 것으로 X축 숫자 1에서 50까지는 7개의 개체가 전송 주기는 2 ms로 했을 경우의 priority 가 가장 높은 개체 1의 전송 지연 시간을 나타내고, 51에서 100까지는 10개의 개체를 사용할 때의 전송 지연 시간을 보여주며, 101에서 150까지는 DPQ Mode를 적용한 경우의 지연 시간을 나타낸다.

개체 수가 늘어날수록 최우선 순위의 개체에 대한 전송 지연 시간도 점진적으로 증가하는 것을 알 수 있으며, DPQ Mode의 적용에 의하여 하위 순위의 개체의 전송을 배려하면

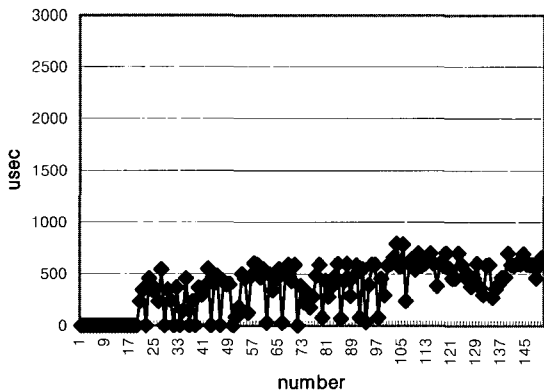


그림 8. 실험 1의 개체 1 (Portable ECU)의 전송 지연. Fig. 8. Transmission delay time of node 1 in experiment 1.

서 최우선 순위의 개체의 전송 시간은 점진적으로 길어지는 것을 알 수 있다.

그리고 그림 12는 실험 2의 경우에 개체 8의 전송지연 시간을 나타낸 것이다. X축으로 1에서 50까지는 개체 수가 10 일 때의 전송지연 시간을 나타내며, 51에서 100 까지는 DPQ Mode를 적용했을 때의 지연 시간을 나타내는 것으로 전송지연이 확연히 줄어들음을 알 수 있다.

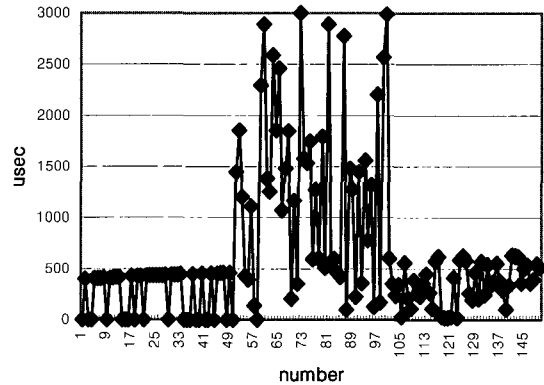


그림 9. 실험 1의 개체 8의 전송지연시간. Fig. 9. Transmission delay time of node 8 in experiment 1.

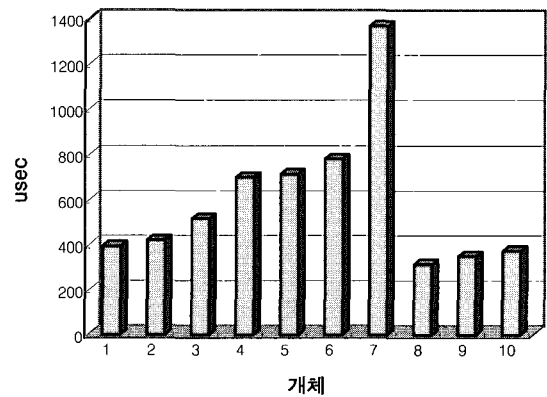


그림 10. 실험 1의 DPQ Mode시 평균 전송지연시간. Fig. 10. Average transmission delay time of DPQ mode in experiment 1.

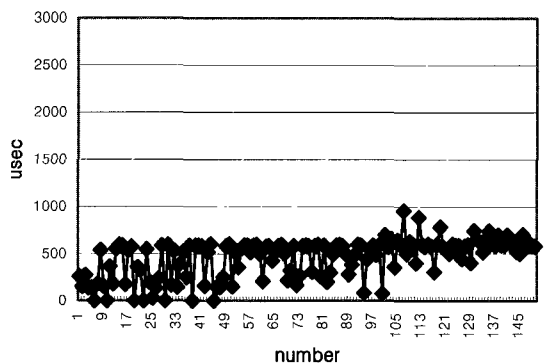


그림 11. 실험 2의 개체 1 (Portable ECU)의 전송지연시간. Fig. 11. Transmission delay time of node 1 in experiment 2.

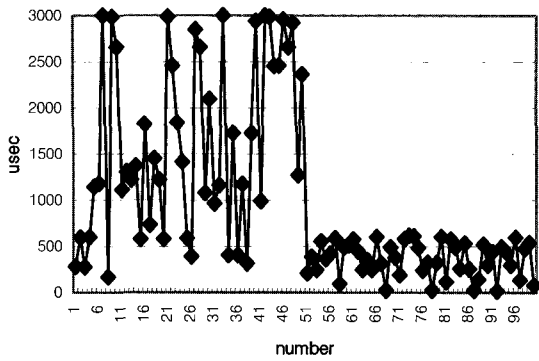


그림 12. 실험 2의 개체 8의 전송지연시간.  
 Fig. 12. Transmission delay time of node 8 in experiment 2.

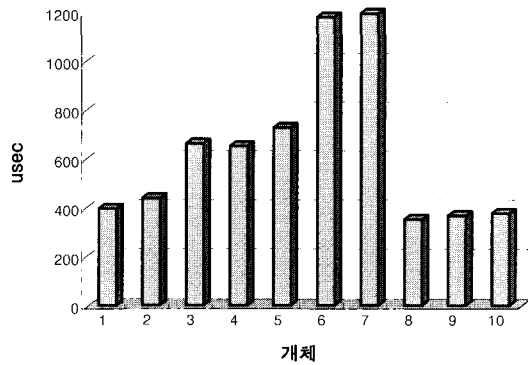


그림 13. 실험 2의 DPQ Mode시 평균 전송지연시간.  
 Fig. 13. Average transmission delay time of DPQ mode in experiment 2.

그림 13은 DPQ Mode를 적용했을 때의 개체들의 평균지연 시간을 나타내는 것으로 DPQ Mode의 적용을 받는 개체 8, 9, 10이 전송지연시간이 가장 짧다는 것을 알 수 있다. 그러나, 이 DPQ Mode의 적용은 동적으로 개체 8의 허용 최대 지연 시간문제가 해결되면 즉시 해제되어 원 priority에 기반하여 전송이 이루어진다는 것을 주목해야 한다.

**VI. 결론**

본 논문은 표준 CAN 프로토콜을 사용하는 네트워크 상에서 일어나는 충돌을 조정하고 고정된 priority 기법에 의해 발생하는 비효율성을 개선하기 위해서 DPQ 기법을 적용하였다. 제시되는 기법은 두 가지의 다른 상황에서의 실험을 통하여 그 유용성을 입증하였다. 첫 번째 실험은 정해진 개체들 사이에서 각각의 개체의 전송주기를 빠르게 하여 전송시 충돌이 자주 발생하여 과부하가 일어나도록 하여 이 때 priority가 낮은 개체의 전송이 DPQ 기법의 적용에 의하여 최대허용 지연시간을 초과하지 않도록 할 수 있음을 보였다.

두 번째 실험은 일정한 전송주기를 가지는 개체의 수를 조절하여 개체의 수를 늘림으로써 전송시 충돌이 자주 발생하게 하여 과부하가 일어나도록 하였으며 전체 시스템의 과부하시에 상대적으로 priority가 낮은 개체에 DPQ 기법을 적용하면 순간적으로 전송 ID의 priority가 상대적으로 높아져서 전송 기회가 보장이 되고 전송 지연 시간이 짧아지는 현상을 실험을 통해 확인하였다.

그러나 우선순위가 높은 개체에 DPQ 기법을 적용했을 경우 표준 CAN을 적용했을 때의 전송시간 보다 개선되는 효과를 보기가 어려웠다. 향후 과제로 동적으로 DPQ 기법을 적용하여 각 개체의 시간지연을 효율적으로 관리하는 알고리즘의 개발할 것이며 이 개발된 알고리즘을 다른 CAN 응용 시스템과의 호환에 있어서 얼마나 편리하게 적용할 수 있는가를 연구할 것이다.

**참고문헌**

- [1] International Standard Organization, "Road - vehicles interchange of digital information - controller area network for high-speed communication" *ISO 11898*, November, 1993.
- [2] 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.
- [3] CAN in AUTOMATION International Users and Manufacturers Group e. V. "CAN application layer (CAL)", CiA/DS201-CiA/DS205, CiA/DS207.
- [4] 박진우, 노동규, 박재한, 허화라, 이장명, "CAN을 이용한 분산 제어 구조를 가지는 이동 로봇 구현" 추계 합동 학술논문 발표회 논문집, pp. 251-255, 1999.
- [5] 홍성수, "분산 실시간 제어 시스템 개발," ICASE 기술 특집: 실시간 제어 시스템(3), *ICASE*, 1, 1998.
- [6] 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.
- [7] K. Tindell and 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.
- [8] K. Tindell and A. Burns, "Guaranteeing message latencies on control area network (CAN)", in *Proc. 1st International CAN Conference*, Mainz, Germany, September, 1994.
- [9] K. Tendell, 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.
- [10] "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 석사과정재학중. 관심분야는 로봇제어를 위한 최적 알고리즘개발, CAN Network, Teleoperation.



### 이 장 명

서울대학교 전자공학과 졸업(1980), 동 대학원 졸업(1982), USC (남가주대학교) 공학박사(1990), 1992년~현 부산대학교 정교수. 주 관심분야는 지능로봇 시스템의 설계 및 제어, 마이크로 프로세서 응용, 시스템 설계, 모터 구동 제어.