

# 노드 기반 스케줄링 방법을 이용한 FlexRay 네트워크 시스템의 구현

김 만 호<sup>1)</sup> · 하 경 남<sup>2)</sup> · 이 석<sup>\*2)</sup> · 이 경 창<sup>3)</sup>

대구경북과학기술원 공공원천기술연구센터<sup>1)</sup> · 부산대학교 기계공학부<sup>2)</sup> · 부경대학교 제어계측공학과<sup>3)</sup>

## Implementation of FlexRay Network System using Node-based Scheduling Method

Man Ho Kim<sup>1)</sup> · Kyoung Nam Ha<sup>2)</sup> · Suk Lee<sup>\*2)</sup> · Kyung Chang Lee<sup>3)</sup>

<sup>1)</sup>Public & Original Technology Research Center, DGIST, Daegu 704-230, Korea

<sup>2)</sup>School of Mechanical Engineering, Pusan National University, Busan 609-735, Korea

<sup>3)</sup>Department of Control and Instrumentation Engineering, Pukyong National University, Busan 608-739, Korea

(Received 4 May 2009 / Accepted 2 September 2009)

**Abstract** : As vehicles become intelligent for convenience and safety of drivers, in-vehicle networking (IVN) systems are essential components of intelligent vehicles. Recently, the chassis networking system which require increased network capacity and real-time capability is being developed to expand the application area of IVN systems. Also, FlexRay has been developed for the chassis networking system. However, FlexRay needs a complex scheduling method of static segment, which is a barrier for implementing the chassis networking system. Especially, if we want to migrate from CAN network to FlexRay network using CAN message database that was well constructed for the chassis networking system by automotive vendors, a novel scheduling method is necessary to be able to reduce design complexity. This paper presents a node-based scheduling method for FlexRay network system. And, in order to demonstrate the method's feasibility, its performance is evaluated through an experimental testbed.

**Key words** : Intelligent vehicle(지능형 자동차), In-vehicle networking system(차량 네트워크 시스템), FlexRay network(플렉스레이 네트워크), Node-based scheduling method(노드 기반 스케줄링 방법), Design complexity(디자인 복잡도)

### 1. 서 론

최근 들어, 자동차의 안정성과 신뢰성에 대한 운전자들의 욕구가 증대됨에 따라,<sup>1-3)</sup> 고속 전송 속도와 예측 가능한(predictable) 전송 특성을 가진 프로토콜이 필요하게 되었다. 학술적으로, Controller Area Network(CAN)<sup>4)</sup>은 속도가 낮고 전송 지연이 비확정적(non-deterministic)이기 때문에, 브레이크나 조향 시스템과 같은 실시간 특성을 요구하는 차시 네트워크(chassis network)용 프로토콜로 적합하지

않다고 알려져 있다.<sup>5)</sup>

이러한 문제를 해결하기 위하여, 새시 제어용 네트워크로 FlexRay가 개발되었으며, 자동차 완성차 업체를 중심으로 그 적용이 확대되어 가고 있는 추세이다. 그러나 FlexRay는 프로토콜 자체의 복잡성 때문에 응용(application)의 구현이 CAN에 비하여 어렵다는 문제점을 가지고 있다. CAN에서는 수년간의 개발과 적용 과정에 따라 다양한 개발 방법론들이 제시되어 있다. 또한, 각 ECU에서 생성되는 센서나 모터 신호의 원활한 전송을 위하여, 완성차업체에 의하여 CAN 메시지 표준이 잘 정립되어 있다.

\*Corresponding author, E-mail: slee@pnu.edu

특히, CAN은 수많은 차량의 새시 네트워크 시스템에 적용되어 안정적으로 동작되고 있다.<sup>6)</sup>

새시 네트워크 시스템에서 CAN 네트워크를 FlexRay 네트워크로 변경하기 위해서는 시스템 설계 단계에서 네트워크의 구성과 동작에 관한 스케줄링이 필요하다. 특히, FlexRay는 기존의 CAN이 지원하지 못하는 기능들을 수행해야 하기 때문에, 네트워크 구성 변수(configuration parameter)와 설정 단계가 더 많이 필요하다는 문제점이 있다. 뿐만 아니라, CAN에 비해 스케줄링이 중요함에도 불구하고 체계적인 스케줄링 방법이나 노하우(know-how)가 없다는 점은 FlexRay를 새시 네트워크 시스템에 적용되는 데 있어 가장 큰 문제로 대두되고 있다.

최근 들어, 이러한 문제를 해결하기 위하여 FlexRay에 대한 다양한 연구들이 발표되었다. Schmidt<sup>7)</sup>는 FlexRay의 정적 구간(static segment)과 동적 구간(dynamic segment)에서의 메시지 스케줄링 방법에 대하여 제안하였으며, Armengaud<sup>8)</sup>는 FlexRay의 동적 구간에서 CAN ID의 우선순위에 따라 슬롯(slot)을 부여하는 방법을 제시하였다. 특히, Seo<sup>9)</sup>는 FlexRay와 CAN을 혼용하기 위한 스케줄링 방법을 제안하였으며, Sethna<sup>10)</sup>는 FlexRay의 구현은 기존 CAN 시스템의 변환으로부터 시작하는 것이 유용하다고 언급하였다. 뿐만 아니라, Coummings<sup>11)</sup>는 차량용 네트워크에서 데이터 전송량이 늘어나면, 게이트웨이(gateway)를 사용한 두 개의 CAN 네트워크 보다는 하나의 FlexRay 네트워크로 대체하여 설계하는 것이 더 효과적이라고 주장하였다.

본 논문은 CAN 네트워크에서 사용되고 있는 메시지를 FlexRay 네트워크의 정적 구간으로 이전(migration)하려고 할 때, FlexRay 네트워크 시스템에서 스케줄링의 복잡성을 줄일 수 있는 노드 기반 스케줄링 방법에 대하여 제안한다. 본 논문에서 제안된 노드 기반 스케줄링 방법에서, 한 노드에서 전송하고자 하는 여러 개의 CAN 메시지들은 하나의 FlexRay 프레임으로 묶여 전송된다. 즉, 발신자 노드는 모든 CAN 메시지를 하나의 FlexRay 프레임으로 묶어 방송(broadcast)하게 되고, 다른 노드들은 방송된 FlexRay 프레임을 수신한 후 자기에게 필요한 CAN 메시지만 추출하여 응용 계층으로 전달하게

된다. 이러한 방법에 의하여, 네트워크 개발자에 의하여 클러스터나 노드 구성 변수가 시스템 설계 시에 오프라인(off-line) 상태에서 정의가 된다면, ECU의 기능 대다수를 담당하는 응용 개발자는 CAN 메시지 데이터베이스만을 참조하여 ECU의 응용 프로그램을 개발할 수 있게 됨으로서 FlexRay 네트워크에 독립적인 개발이 가능하게 될 것이다.

본 논문은 서론을 포함하여 5장으로 구성되어 있다. 2장에서는 FlexRay에 대하여 간략하게 소개하고, 3장에서는 FlexRay 네트워크의 노드 기반 스케줄링 방법을 제안한다. 4장에서는 FlexRay 네트워크에서 노드 기반 스케줄링 방법의 구현 방법을 제시하고, 그 사용 가능성을 평가한다. 마지막으로, 5장에서 결론 및 향후 연구 방향에 대해서 서술한다.

## 2. FlexRay 개요

FlexRay<sup>12)</sup>는 BMW와 Daimler Chrysler, Motorola, Philips 등이 참여하는 FlexRay 컨소시엄에 의하여 개발된 분산 실시간 차량용 네트워크 프로토콜이다. FlexRay는 10Mbps의 고속 전송 속도를 지원하며, 듀얼 채널(dual channel)을 사용함에 따라 중복성(redundancy)을 제공한다. 또한, 전송 지연이 확정적(deterministic)이고 예측 가능(predictable)하다는 특징을 가지고 있다.

FlexRay에서 통신 사이클(communication cycle)은 Fig. 1(a)와 같이 동기 프레임 전송이 가능한 정적 구간(static segment)과 비동기 프레임 전송이 가능한 동적 구간(dynamic segment), 프로토콜에 정의된 심벌(symbol)을 전송할 수 있는 심벌 윈도우(symbol window), 네트워크에서 시간 동기화를 위한 네트워크 유휴 시간(network idle time)으로 구성되어 있다.

Fig. 1(b)는 정적 구간에서 FlexRay의 매체 접속 제어 방법을 나타낸다. 정적 구간에서, FlexRay는 TDMA(time division multiple access) 방식을 이용하여 프레임을 전송한다. 그림에서, 정적 구간은 같은 길이를 가지는 정적 슬롯(static slot)으로 나누어진 다. 각 노드들은 시스템 설계 시에 채널별로 스케줄링을 수행한다. 네트워크가 시작되면, 각 노드는 자신이 가진 프레임 ID(frame ID)와 같은 슬롯 카운터(slot counter) 값을 가지는 슬롯에 해당 프레임을 삽

입하여 전송한다. 이 때, 각 노드는 프레임의 고정 허용(fail-safe)을 위하여 두 개의 채널에 같은 프레임 ID를 가지는 프레임을 전송할 수도 있다.

Fig. 1(c)는 동적 구간에서 FlexRay의 매체 접속 제어 방법을 나타낸다. 동적 구간에서, FlexRay는 동적 미니 슬롯팅(dynamic mini-slotting)이라는 방법을 사용한다. 그림에서, 동적 구간은 매우 작은 길이를 가지는 미니 슬롯(mini slot)으로 나누어진다. 여기에서, 슬롯 카운터는 정적 구간의 마지막 슬롯 번호보다 1 큰 값으로 설정된다. 동적 구간에서, 프레임 전송 방법은 정적 노드의 프레임 전송 방법과 유사하다. 즉, 각 노드에서 생성되는 프레임은 프레임 ID를 가지고 있으며, 각 노드는 프레임 ID와 동일한 슬롯 카운터 값을 가지는 미니 슬롯에 해당 프레임을 삽입하여 전송한다. 다만, 미니 슬롯의 개수는 프

레이의 길이의 크기에 따라 동적으로 할당된다. 즉, 그림에서, 프레임 ID  $m$ 은 5 내지 6개의 미니 슬롯 길이를 가지기 때문에, 6개의 미니 슬롯이 할당된다. 이 때, 다음에 나오는 7번째 미니 슬롯의 슬롯 카운터는  $m+1$ 로 설정되고, 프레임 ID가  $m+1$ 인 프레임이 해당 미니 슬롯에서 전송될 수 있다.<sup>12)</sup>

### 3. 노드 기반 스케줄링 방법을 이용한 FlexRay 네트워크 시스템의 설계

일반적으로, FlexRay 네트워크 시스템의 스케줄링은 다음과 같은 두 단계로 이루어진다. 첫 번째 단계는 네트워크 시스템의 전반적인 구조를 결정하는 클러스터(cluster) 설계이며, 두 번째 단계는 각 노드의 세부적인 기능을 설계하는 노드(node) 설계이다. 먼저, 클러스터 설계 단계에서는 CAN 정보를 바탕

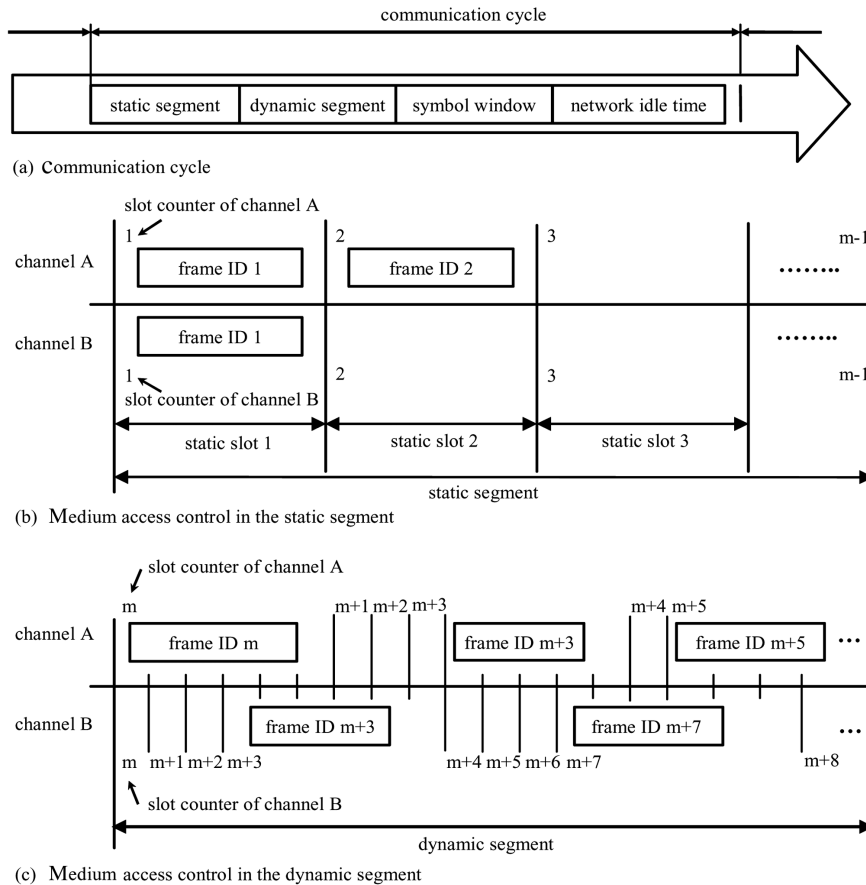


Fig. 1 Communication cycle and medium access control method of FlexRay

으로 네트워크의 전송 속도, 통신 사이클의 길이, 프레임의 크기와 정적 구간의 크기와 같은 파라미터가 설정된다. 다음으로, 노드 설계 단계에서는 통신 채널, 프레임 ID와 메시지 기능과 같은 파라미터가 설정된다. 이러한 스케줄링은 오프라인으로 이루어진다.

그러나 FlexRay 네트워크 시스템에서 메시지 개수가 늘어나면, 할당해야 하는 정적 슬롯의 개수가 증가하고 노드 설계 단계에서의 작업량이 증가하게 된다. 특히, CAN 네트워크 시스템에서 정의되어 있는 메시지는 수백 개이기 때문에, CAN 메시지를 FlexRay 네트워크의 정적 슬롯으로 이전하기 위한 스케줄링 작업은 매우 복잡하고 방대하다. 뿐만 아니라, FlexRay 네트워크 시스템에서 오프라인으로 스케줄링이 결정되고 나면, 노드의 추가나 메시지의 추가가 매우 어렵다. 일반적으로, 노드나 메시지를 추가하려면, 네트워크가 중단되어야 할 뿐만 아니라 모든 노드의 스케줄링을 바꾸어야 한다는 문제점이 있다.

따라서 본 논문에서는 CAN 네트워크 시스템에서 정의된 메시지들을 FlexRay 네트워크로 효율적으로 이전할 수 있는 Fig. 2와 같은 노드 기반 스케줄링 방법을 제안한다. 그림에서, FlexRay의 정적 구간의 슬롯은 각 노드에 일대일로 할당되어 있다. 즉,

정적 구간의 슬롯의 개수는 노드의 개수와 동일하다. 발신자 노드는 통신 사이클 동안에 생성되어 있는 CAN 메시지의 ID와 데이터들을 모아서 하나의 FlexRay 프레임으로 조립(assembly)하여 방송한다. 여기에서, FlexRay 프레임의 페이로드(payload)는 n개의 SMD(start message delimiter), ID(identifier), 메시지 쌍과 슬롯 처리 시간(slot processing time)으로 구성되어 있다. SMD는 페이로드에서 CAN 메시지를 구분하기 위한 3비트의 지시자이며, ID는 CAN 2.0B의 29비트 식별자이다. 메시지는 CAN의 최대 메시지 길이인 8byte로서 고정된 길이를 가진다. 마지막으로, 슬롯 처리 시간은 각 노드들이 슬롯을 수신하고 난 후, 슬롯으로부터 자신에게 해당되는 CAN 메시지만을 추출해내는 데 걸리는 시간이다.

노드 기반 스케줄링 방법에서, 다른 노드들은 해당 슬롯에 들어있는 프레임을 모두 읽을 수 있어야 하며, 프레임으로부터 자기가 수신자로 지정된 CAN 메시지만 추출할 수 있어야 한다. 이러한 방법에 따라, FlexRay의 스케줄링은 각 노드에 슬롯을 할당하는 단순한 작업으로 변경되게 된다. 특히, CAN 메시지의 추가나 삭제에도 FlexRay의 스케줄링은 변경할 필요 없게 된다. 즉, CAN 메시지의 발신자 노드와 수신자 노드의 응용 프로그램만 수정하면 통신이 이루어질 수 있으므로, FlexRay 네트워크의

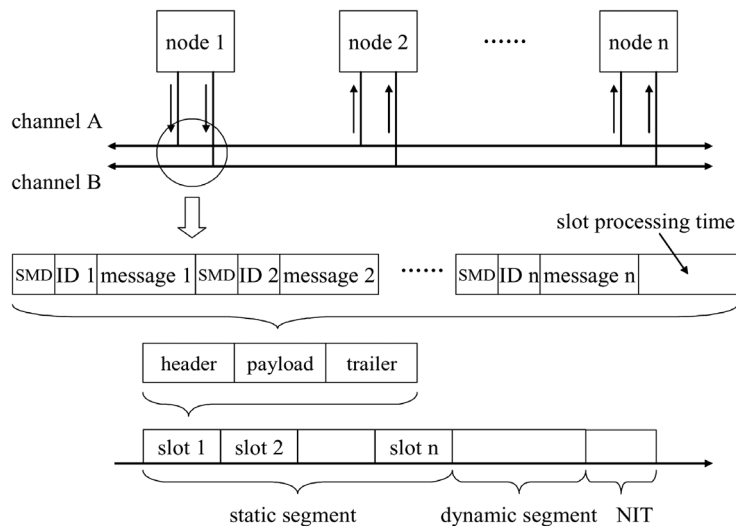


Fig. 2 Communication structure of FlexRay network for node-based scheduling method

스케줄링이나 다른 노드들의 응용 프로그램은 변경할 필요가 없게 된다.

그러나 FlexRay 노드에서 CAN 메시지를 하나의 FlexRay 프레임으로 조립하고, FlexRay 프레임으로부터 CAN 메시지를 추출해 내기 위한 기능이 필요하다. Fig. 3은 노드 기반 스케줄링 방법을 위한 FlexRay 노드의 계층 구조를 나타내고 있다. 일반적으로, FlexRay 노드는 물리 계층, 데이터 링크 계층, 전달 계층, 응용 계층으로 이루어져 있다. 그러나 노드 기반 스케줄링 방법에서는 슬롯 관리 부계층(slot management sublayer)이 응용 계층(application layer)에 추가된다. 슬롯 관리 부계층은 슬롯 조립(slot assembly), 슬롯 분해(slot disassembly), 메시지 필터링(message filtering)과 같은 3가지 기능을 가진다. 먼저, 슬롯 조립 기능은 응용 계층으로부터 받은 CAN 메시지에서 ID와 데이터를 추출하여 하나의 FlexRay 프레임으로 만드는 작업이다. 다음으로, 슬롯 분해 기능은 하위 계층으로부터 받은 FlexRay 프레임으로부터 ID와 데이터를 추출하여 CAN 메시지로 만드는 작업이다. 마지막으로, 메시지 필터링 기능은 수신된 CAN 메시지로부터 자신이 수신자

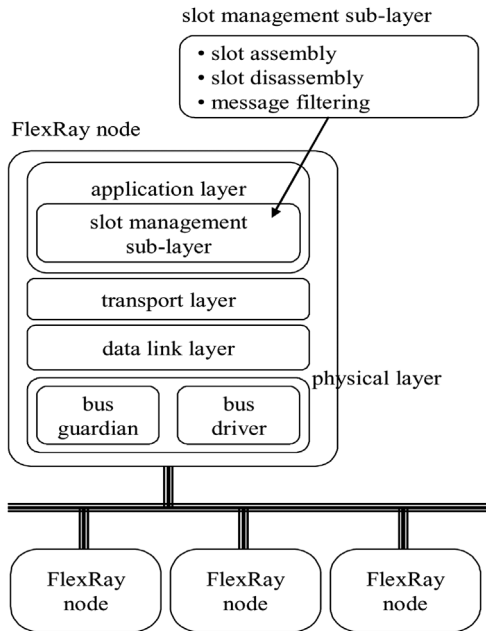


Fig. 3 Layer architecture of FlexRay node for node-based scheduling method

로 지정된 메시지는 수신하고, 그렇지 않은 메시지는 무시하는 작업이다.

슬롯 조립 단계에서, CAN 메시지가 큰 전송 지연(time delay)이나 큰 전송 지터(jitter)없이 전송되기 위해서는, FlexRay 네트워크의 통신 사이클이 네트워크에서 생성되는 CAN 메시지들의 생성 주기의 최소공약수로 설정되어야 한다. 일반적으로, CAN 네트워크에서 메시지의 발생 주기는 수 msec인데 반하여, FlexRay의 통신 사이클은 1msec 이내로 설정하더라도 큰 문제없이 메시지를 전송할 수 있다. Fig. 4는 CAN 메시지의 생성 주기와 FlexRay 프레임의 전송 주기에 관련된 타이밍 도(timing diagram)의 예를 나타내고 있다. 그림에서, 발신 노드 x에서 CAN 메시지 1과 3의 생성 주기는 FlexRay 통신 사이클과 같고, 메시지 2의 생성 주기는 통신 사이클의 주기의 두 배라고 하자. 그림에서, FlexRay 통신 사이클을 CAN 메시지의 생성 주기의 최대공약수로 설정하게 되면, m1과 m3은 매 통신 사이클마다 슬롯 x에서 항상 전송이 될 것이고, m2는 번갈아가면서 슬롯 x에서 전송될 것이다. 또한, CAN 메시지가 생성되고 나면 다음 통신 사이클에서 전송될 것

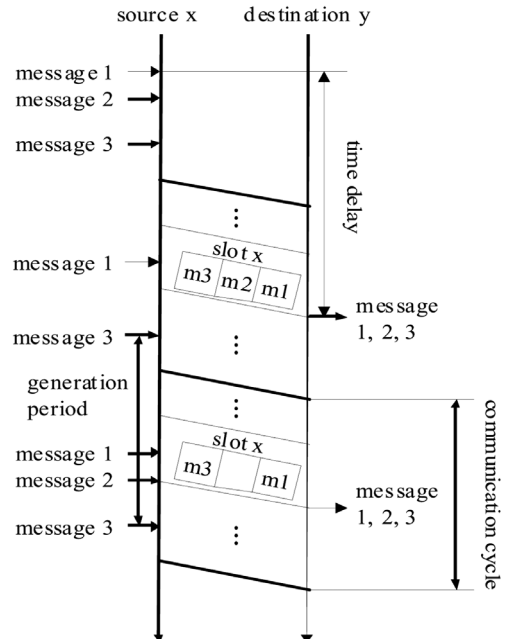


Fig. 4 Example of timing diagram of FlexRay node for node-based scheduling method

이기 때문에, CAN 메시지의 전송 지연은 1 통신 사이클 이내가 될 것이다. 또한, 생성된 CAN 메시지는 FlexRay 네트워크에서 슬롯 x를 이용하여 주기적으로 보내어질 것이기 때문에, 전송 지연은 거의 발생하지 않을 것이다.

#### 4. 노드 기반 스케줄링 방법을 이용한 FlexRay 네트워크 설계

본 절에서는 노드 기반 스케줄링 방법의 사용 가능성을 평가하기 위하여, FlexRay 네트워크의 실험 모델을 구현하고 스케줄링 방법의 성능을 검토하였다. 먼저, 본 논문에서는 Fig. 5와 같은 실험 모델을 구축하였다. FlexRay 네트워크 실험 모델에는 TZM사의 FlexNode 세 개와 CANoe.FlexRay V7.0이 접속되어 있다. 여기에서, FlexNode의 클러스터 설계와 노드 설계는 TZM사의 FlexConfig를 이용하여 수행하였다. 또한, FlexRay 노드에서 송수신되는 프레임을 모니터링하기 위하여, CANoe.FlexRay의 FIBEX(field bus exchange format) 파일을 이용하여 가상의 노드를 구현하였다.

FlexRay 네트워크에는 Table 1과 같이 8byte 길이를 가지는 CAN 메시지가 생성된다고 가정한 후, 스케줄링을 수행하였다. 스케줄링은 두 가지 방법으로 구현하였다. 첫 번째 방법은 한 개의 CAN 메시지를 한 개의 슬롯을 이용하여 전송하는 방법을 사용하였다. 본 논문에서는 이러한 방법을 메시지 기반 스케줄링 방법(message-based scheduling method)이라고 정의하였다. 두 번째 방법은 본 논문에서 제안한 노드 기반 스케줄링 방법을 사용하였다.

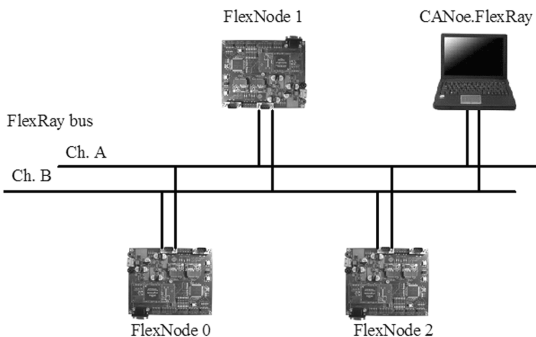


Fig. 5 Testbed of FlexRay network with node-based scheduling method

Table 1 Example of CAN message

Message number	CAN message ID	Tx	Rx	Period (msec)
MSG_1	0x001	0	1	5
MSG_2	0x004	0	2	10
MSG_3	0x007	0	1,2	10
MSG_4	0x009	0	2	20
MSG_5	0x00c	0	1	40
MSG_6	0x003	1	0,2	5
MSG_7	0x005	1	0	10
MSG_8	0x008	1	0,2	20
MSG_9	0x00a	1	0	40
MSG_10	0x00d	1	2	80
MSG_11	0x002	2	1	5
MSG_12	0x006	2	0	10
MSG_13	0x00b	2	1	40
MSG_14	0x00e	2	0	80
MSG_15	0x00f	2	1	160

- 메시지 기반 스케줄링 방법에서 클러스터 구성 변수는 다음과 같이 설정된다.
  - 데이터 전송 속도는 10Mbps로 설정하고, 전송 채널은 한 개만을 사용한다.
  - 세 개의 노드가 각각 5개의 메시지를 개별 정적 슬롯을 사용하여 전송한다.
  - 통신 사이클 길이는 메시지들의 생성 주기의 최대공약수인 5msec로 설정한다.
  - FlexRay 페이로드에는 FlexRay 표준에서 규정된 바에 따라 17.2MT로 설정된다. 또한, 정적 슬롯의 크기는 27MT(macrotick)로 설정된다. 여기에서, MT는 FlexRay 표준에서 정의되는 시간으로서, 10비트 타임(bit time)이다. 다음으로, 노드 구성 변수는 다음과 같이 설정된다.
  - 각 노드에서 주기가 가장 짧은 5msec로 전송되는 메시지를 싱크 프레임(sync frame)으로 설정한다.
  - 세 노드가 매 사이클마다 모두 한 개씩 싱크 프레임을 전송하도록 설정한다.
- 이러한 구성을 바탕으로 FlexRay 네트워크의 스케줄링을 수행하면 Table 2와 같은 결과를 얻을 수 있다. FlexRay 네트워크에서, 통신 사이클의 길이는 5msec, 즉 5000MT이다. 또한, 27MT 길이의 정적 슬롯 15개가 들어가는 정적 구간의 길이는 405MT이

Table 2 Message-based scheduling for FlexRay network

Slot ID	Message number	Tx	Rx	Period (msec)
1	MSG_1	0	1	5
2	MSG_11	2	1	5
3	MSG_6	1	0, 2	5
4	MSG_2	0	2	10
5	MSG_7	1	0	10
6	MSG_12	2	0	10
7	MSG_3	0	1, 2	10
8	MSG_8	1	0, 2	20
9	MSG_4	0	2	20
10	MSG_9	1	0	40
11	MSG_13	2	1	40
12	MSG_5	0	1	40
13	MSG_10	1	2	80
14	MSG_14	2	0	80
15	MSG_15	2	1	160

Table 3 Node-based scheduling for FlexRay network

Slot ID	Message number	Tx	Rx	Period (msec)
1	MSG_1	0	0,1,2	5
	MSG_2	0	0,1,2	10
	MSG_3	0	0,1,2	10
	MSG_4	0	0,1,2	20
	MSG_5	0	0,1,2	40
2	MSG_6	1	0,1,2	5
	MSG_7	1	0,1,2	10
	MSG_8	1	0,1,2	20
	MSG_9	1	0,1,2	40
	MSG_10	1	0,1,2	80
3	MSG_11	2	0,1,2	5
	MSG_12	2	0,1,2	10
	MSG_13	2	0,1,2	40
	MSG_14	2	0,1,2	80
	MSG_15	2	0,1,2	160

다. 나머지 4595MT는 NIT로 설정된다.

노드 기반 스케줄링 방법을 이용하는 경우, 클러스터 구성 변수는 다음과 같이 변경될 수 있다.

- 세 노드는 전송하려는 5개의 메시지를 한 개의 FlexRay 프레임으로 조립하여 정적 슬롯 하나만 사용한다.
- FlexRay 페이로드 크기는 5개의 CAN 메시지가 전송될 수 있도록 69.2MT로 설정되고, 정적 슬롯의 크기는 80MT로 설정된다.

다음으로, 노드 구성 변수는 다음과 같이 설정된다.

- 모든 노드가 한 통신 사이클에서 한 프레임만 전송하므로 FlexRay 프레임에 싱크 정보를 실어서 전송한다.
- 모든 노드는 모든 FlexRay 프레임을 수신한 후, 슬롯 관리 부계층에서 정의되어 있는 CAN 메시지 규격에 따라 필요한 메시지만 추출하여 수신한다.

이러한 구성을 이용하여 정적 슬롯을 배정하고 스케줄링을 수행하면 Table 3과 같은 결과를 얻을 수 있다. 통신 사이클의 길이는 5msec이며, 80MT 길이의 정적 슬롯 3개가 들어가는 정적 구간의 길이는 240MT이다. 나머지 4760MT는 NIT로 설정된다.

Fig. 6은 노드 기반 스케줄링 방법을 이용하여 스케줄링된 FlexNode 0에서 송신되는 메시지들을 CANoe.FlexRay를 이용하여 모니터링한 결과를 나타내고 있다. 그림에서, FlexNode 0으로부터 Table 3과 같은 주기를 가진 메시지들이 전송 지터 없이 정상적으로 전송되는 것을 확인할 수 있다.

메시지 기반 스케줄링 방법에서 정적 구간의 길이는 405MT임에 반하여, 노드 기반 스케줄링 방법에서는 약 60%의 정적 슬롯 길이만으로 전송이 가능하게 된다. 특히, 한 개의 정적 슬롯에서 15개 정도의 CAN 메시지가 전송될 수 있도록 설계하는 경우, 페이로드 크기는 189.2MT, 정적 슬롯 크기는 200MT가 되며, 정적 구간의 길이는 600MT가 된다. 이러한 구성을 하게 되면, FlexRay 노드에서 CAN 메시지가 추가되더라도 오프라인에서 수행되어야 하는 스케줄링의 변경 없이 해당 노드의 슬롯 관리 부계층의 응용 프로그램만 변경함으로써, 효과적인 스케줄링이 가능해지게 된다.

## 5. 결론

본 논문에서는 CAN 메시지 데이터베이스를 이용하여 FlexRay 네트워크 시스템을 설계할 때, 스케줄링의 복잡도를 줄이기 위한 방법으로 FlexRay 네트워크의 노드 기반 스케줄링 방법에 대하여 제

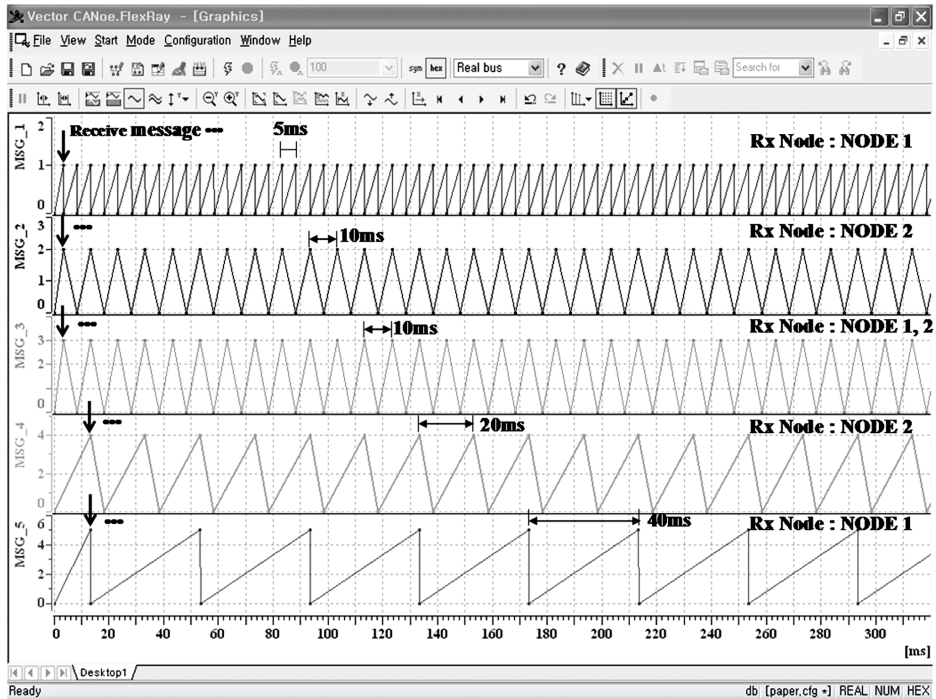


Fig. 6 Experimental result node message of static segment using r node based scheduling method

안하였다. 이를 위하여, 본 논문에서는 특정 노드에서 생성되는 CAN 메시지들을 하나의 FlexRay 슬롯을 이용하여 전송할 수 있는 FlexRay 네트워크 설계 방법을 제시하였다. 마지막으로, FlexRay 네트워크 시스템 실험 모델을 구현하고, 제안된 노드 기반 스케줄링의 사용 가능성을 살펴보았다. 이상과 같은 연구 결과로부터 다음과 같은 결론을 얻을 수 있었다.

- 1) FlexRay 네트워크 시스템의 설계는 클러스터 설계와 노드 설계로 이루어진다. 만약, 자동차 완성차업계에서 사용되고 있는 CAN 메시지 데이터베이스를 이용하여 FlexRay 네트워크를 설계하는 경우, 효과적인 FlexRay 네트워크 설계가 가능함을 알 수 있다.
- 2) 노드 기반 스케줄링 방법을 이용하는 경우, 클러스터나 노드 구성, 네트워크의 동작과 관련된 기능들이 네트워크 개발자에 의하여 시스템 설계시에 오프라인 상태에서 정의되고 나면, ECU의 기능 대다수를 담당하는 응용 개발자는 CAN 메시지 데이터베이스만을 참조하여 ECU를 개발

할 수 있게 됨으로서 FlexRay 네트워크에 독립적인 개발이 가능하게 될 것이다.

- 3) 노드 기반 스케줄링 방법에서 각 슬롯의 길이를 충분히 길게 함으로써, FlexRay 노드에서 CAN 메시지가 추가되더라도 오프라인에서 수행되어야 하는 스케줄링의 변경 없이 ECU의 응용 프로그램만의 변경에 의하여 효과적인 스케줄링이 가능하게 될 것이다.

그러나 본 논문은 실험실 수준의 실험 모델을 이용하여 한정된 메시지 발생 조건에서 노드 기반 스케줄링 기법의 적용 가능성만을 검증하였다. 이러한 스케줄링 방법이 자동차에 적용되기 위해서는 새시 네트워크 시스템용 CAN 메시지 데이터베이스를 이용한 FlexRay 네트워크 설계에 대한 연구가 필요하다. 또한, 제안된 스케줄링 방법의 성능을 검증할 수 있는 정성적이고 정량적인 평가 방법의 개발이 필요하다. 마지막으로, 노드 기반 스케줄링 방법의 효율성을 높이기 위한 다양한 형태의 추가적인 스케줄링 방법에 대한 연구가 필요하다.



## 후 기

이 논문은 2007년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2007-314-D00022).

## References

- 1) G. Leen and D. Herrernam, "Expending automotive electronic," IEEE Computer, Vol.35, No.1, pp.88-93, 2002.
- 2) A. Broggi, "Intelligent Vehicle Applications Worldwide," IEEE Intelligent Systems, Vol.15, No.1, pp.78-81, 2000.
- 3) T. H. Hwang, K. Park, S. J. Heo, S. H. Lee and J. C. Lee, "Design of Integrated Chassis Control Logics for AFS and ESP," Int. J. Automotive Technology, Vol.9, No.1, pp.17-27, 2008.
- 4) J. Ryu, M. Yoon and M. Sunwoo, "Development of a Network-based Traction Control System, Validation of Its Traction Control Algorithm and Evaluation of Its Performance Using Net-HILS," Int. J. Automotive Technology, Vol.7, No.6, pp.687-695, 2006.
- 5) H. Kopetz, "A Comparison of CAN and TTP," Annual Reviews in Control, Vol.24, No.1, pp.177-188, 2000.
- 6) N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, "Trends in Automotive Communication Systems," Proceedings of the IEEE, Vol.93, No.6, pp.1204-1223, 2005.
- 7) K. Schmidt and E. G. Schmidt, "Message Scheduling for the FlexRay Protocol: The Static Segment," IEEE Transactions on Vehicular Technology, Vol.58, No.5, pp.2170-2179, 2009.
- 8) E. Armengaud, A. Steininger and M. Horauer, "Towards a Systematic Test for Embedded Automotive Communication Systems," IEEE Transactions on Industrial Informatics, Vol.4, No.3, pp.146-155, 2008.
- 9) S. H. Seo, T. Y. Moon, J. H. Kim, S. H. Hwang, and J. W. Jeon, "A Gateway System for an Automotive System: LIN, CAN and FlexRay," 6th IEEE International Conference on Industrial Informatics, pp.967-972, 2008.
- 10) F. Sethna, E. Stipidis and F. H. Ali, "What Lessons Can Controller Area Networks Learn from FlexRay," IEEE Vehicle Power and Propulsion Conference, pp.1-4, 2006.
- 11) R. Cummings, "Easing the Transition of System Designs from CAN to FlexRay," SAE 2008-01-0804, 2008.
- 12) FlexRay Consortium, FlexRay Protocol Specification 2.1 Revision A, 2005.