

## CAN 통신 WCRT 분석을 통한 IEEE1212 기반 RTU의 통신 신뢰성 검증

황호성\*, 엄복진\*

현대중공업 전기전자시스템연구소

### Communication Reliability Test of IEEE1212 Based RTU through WCRT Analysis of CAN

Hyo-Seong Hwang\*, Bok-Jin Youm\*

Electro Electric Systems Research Institute / Hyundai Heavy Industries Co., Ltd.

**Abstract** - 본 논문은 CAN(Controller Area Network) 통신의 WCRT(Worst Case Response Time) 분석을 통해 IEEE1212 기반 RTU(Remote Terminal Unit)의 통신 신뢰성을 검증한다. RTU는 원격지에서 네트워크를 구성하여 각 모듈들로부터 획득한 정보를 서버로 전송하거나 원격지를 제어하는 역할을 한다. 본 논문에서 대상으로 하는 RTU의 네트워크는 물리계층과 데이터링크 계층은 CAN을 사용하고 응용계층은 IEEE1212 기반으로 통신한다. 즉, RTU를 구성하는 모듈은 각각 메모리 영역을 가지고 있고, CAN ID에 메모리 주소를 포함하여 해당 영역을 쓰는 구조로 통신을 한다. 첫 번째로 이러한 응용계층을 고려하여 WCRT를 계산하는 방법을 제시하고, 두 번째로 CAN 메시지의 우선순위를 최적화하는 방법을 고찰한다. 마지막으로 시험 환경을 구축하고 시뮬레이션을 통해 앞에서 선정된 우선순위에 따라 통신 신뢰성을 검증한다.

#### 1. 서 론

IT 융합을 통한 공장, 선박, 플랜트 등의 성능 향상에 대한 관심이 높아지면서, 분산형 시스템을 통한 시스템 통합 및 자동화 솔루션 개발이 가속화되고 있다. 분산형 시스템의 구성요소중 하나인 RTU(Remote Control Unit)는 원격지에서 시스템 혹은 장치들을 제어 및 감시하여 그 결과를 서버에 전송해주는 역할을 한다. 분산형 시스템의 적용범위와 통합해야 하는 시스템들이 증가함에 따라 RTU 내부 네트워크의 트래픽이 점점 증가하고 있고, 통신 신뢰성 또한 중요해지고 있다.

CAN에서의 응답시간 분석을 통한 신뢰성 검증에 대한 연구는 꾸준히 계속되고 있다. [1]에서는 [2]에서 제시한 고정형 우선순위 선점형 프로세서의 분석모델을 이용하여 CAN에서의 응답시간을 분석하였고, 우선순위 최적화를 위해 DM(Deadline Monotonic) 스케줄링 기법을 제시하였다. 하지만 CAN 통신은 우선순위가 높은 패킷도 전송중인 메시지는 중단시키지 못함으로 [3]에서는 비선점형 환경에서의 스케줄링 기법을 제시하였다. 이러한 연구는 CAN 통신 계층인 물리계층과 데이터 링크까지의 분석만 수행하였다.

본 논문에서는 IEEE1212 기반 RTU의 응용 계층까지 함께 고려하여, 통신 신뢰성 검증 방법을 제시한다.

#### 2. 본 론

##### 2.1 IEEE1212 기반 RTU CAN

CAN 통신은 차량 내 ECU(Electronic Control Unit) 간 통신 방법으로 개발되어, 현재는 선박, 플랜트, 공장 등 자동화 영역에 광범위하게 사용되고 있다. 멀티마스터 방식으로 ID를 가지고 브로드 캐스팅하여 통신 하며, CSMA/CR(Carrier Sense Multiple Access/Collision Resolution)을 이용하여 낮은 CAN ID가 채널에 먼저 접근할 수 있도록 매체 접근 제어를 한다. 이러한 CAN 통신의 신뢰성을 검증하기 위해서는 생성되는 메시지의 길이, 주기성, 주기, 우선순위, 데드라인의 ID 특성을 알아야 한다. 본 장에서는 기존 연구의 CAN 통신 응답시간 분석 모델과 IEEE1212 기반 RTU 네트워크 구조에 대해 알아본다.

##### 2.1.1 CAN(Controller Area Network) 통신 WCRT 계산

CAN 통신에서 WCRT는 식(1)과 같이 데이터를 전송하기 위해 기다려야 하는 최대시간  $t_m$ 과 데이터 전송에 걸리는 최대시간  $C_m$ 으로 구성된다[1].  $C_m$ 은 메시지 바이트 수에 따라 다르게 되는데, bit stuffing을 고려하여  $s_m$  바이트의 최대 전송 시간은 식(2)와 같다.  $t_m$ 은 낮은 우선순위 메시지가 이미 채널을 점유하여 생기는 최대 지연시간 B와 높은 우선순위 메시지의 채널 점유로 인한 최대 지연시간으로 구성된다. 전자는 8바이트 전송시간과 같고, 후자는 식(3)을 통해 구할 수 있다.  $J_j$ 는 메시지가 발생하는 지터이고  $\tau_{bit}$ 은 한 비트당 전송시간이다. 식(3)은  $t_m$ 에 대한 식으로 변형할 수 없지만 식(4)와 같이 점화식으로 변형하여 조

기 값을 이용하여  $t_m$ 을 구할 수 있다[1]. 본 논문에서는 식 (1)-(4)[1]를 이용하여 WCRT를 계산한다.

$$WCRT_m = t_m + C_m \quad (1)$$

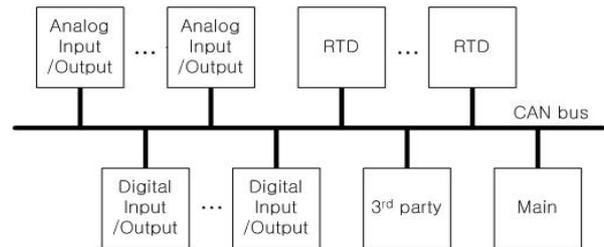
$$C_m = (80 + s_m)\tau_{bit} \quad (2)$$

$$t_m = B + \sum_{\forall j \in hp(m)} \left\lceil \frac{t_m + J_j + \tau_{bit}}{T_j} \right\rceil C_j \quad (3)$$

$$t_m^{n+1} = B + \sum_{\forall j \in hp(m)} \left\lceil \frac{t_m^n + J_j + \tau_{bit}}{T_j} \right\rceil C_j \quad (4)$$

##### 2.1.2 IEEE1212 기반 RTU 종류 및 통신 구조

IEEE1212 마이크로컴퓨터 버스를 위한 통신 프로토콜로 메모리 영역을 가지고 어드레싱하여 읽고 쓰는 방식으로 통신한다. IEEE1212 기반 RTU는 IEEE1212를 간소화하여 각 RTU 모듈마다 16비트 어드레싱이 가능한 주소영역을 가지고 있고 CAN ID에 이 주소를 포함하여 통신한다. 그림 1은 RTU 모듈 종류와 연결된 네트워크를 나타낸다. 메인 모듈은 나머지 모듈들과 통신하여 수집한 정보를 서버로 전송하거나 나머지 모듈들에게 출력 값을 주는 역할을 한다. 표1은 각 모듈들이 메인 모듈과 통신하는 메시지 길이를 나타낸다. 3rd Party의 포인트 길이는 analog인 경우 2bytes이고 digital인 경우 1bit이다.



<그림 1> RTU 종류 및 구조

<표 1> 각 모듈들이 통신하는 메시지 길이

RTU 종류	데이터 길이
Analog Input/Output	채널 수 x 2 bytes
RTD	채널 수 x 2 bytes
Digital Input/Output	채널 수 x 1 bit
3rd Party	채널 수 x 포인트 수 x 포인트 길이

##### 2.2 통신 신뢰성 검증

CAN 통신은 낮은 ID가 높은 우선순위를 가지기 때문에, WCRT를 계산하여 통신 신뢰성을 검증하기 위해서는 각 모듈들에서 생성하는 CAN 메시지 특성을 알아야 한다. 본 장에서는 IEEE 기반 RTU의 응용 계층을 고려하여 통신 WCRT를 분석하고 우선순위를 최적화 한다.

##### 2.2.1 데이터의 ID 구성 및 전송시간

본 논문에서 대상으로 하는 IEEE1212 기반 RTU는 29비트 주소를 사용하는 CAN2.0B를 사용한다. 하위 ID 16비트에 메모리 주소를 포함해야 하기 때문에, 우선순위는 상위 13 비트에 담게 된다. CAN 통신은 한 메시지 당 최대 8바이트를 포함할 수 있으므로, 표1에서 보여주는 각 RTU의 데이터 길이는 8바이트보다 클 경우 N개의 메시지로 나누어져 연속적으로 보내진다. 이때 CAN ID는 메모리영역만 순차적으로 증가하고 우선순위 영역은 동일하다. 즉, 각 모듈들이 생성한 데이터는 여러

CAN 메시지로 나누어지더라도 같은 우선순위 비트로 CSMA/CR을 한다. 그렇기 때문에 응용계층에서의 최대 전송시간  $C_{tm}$ 은 식(5)와 같다.  $S_{im}$ 은 마지막 CAN 메시지에 포함된 바이트의 길이이다.

$$C_{tm} = (160(N-1) + 80 + 10s_{fm})\tau_{bit} \quad (5)$$

### 2.2.2 우선순위 최적화

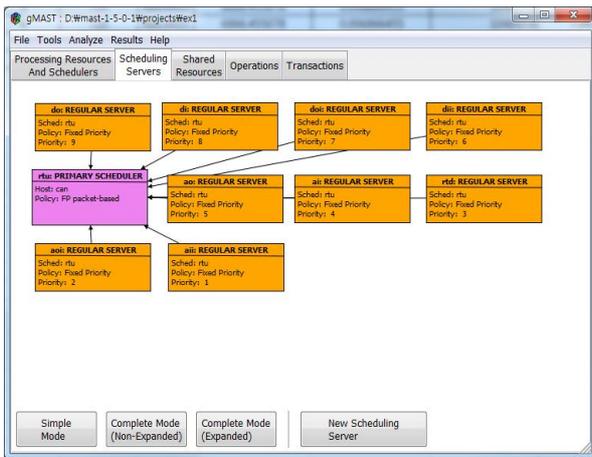
RTU의 각 모듈들은 미리 정의된 업데이트 주기(UP)를 만족하기 위해 주기적으로 통신을 해야 하며, 업데이트 주기를 항상 만족하기 위해서는 WCRT를 고려하여 업데이트 주기가 유효한지 검사해야 한다. 만약 WCRT가 업데이트 주기보다 크다면, 업데이트 주기를 만족하지 못하는 경우가 발생할 수 있다. 즉, 업데이트 주기는 다음 조건을 만족해야 통신 신뢰성을 만족한다.

$$WCRT_m \leq UP_m$$

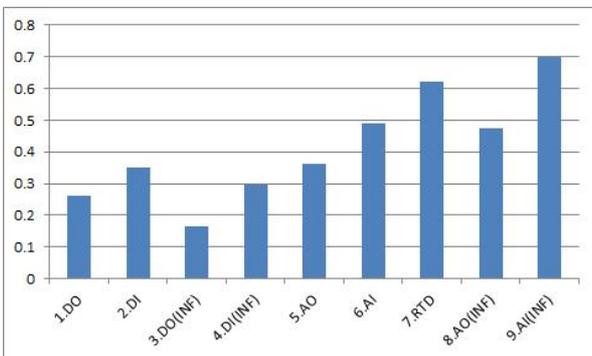
WCRT는 메시지의 우선순위에 따라 다른 값을 가지기 때문에, CAN 네트워크 안에 모든 메시지의 우선순위를 최적화하여 모든 메시지가 통신 신뢰성 조건을 만족하도록 해야 한다. 우선순위를 최적화하는 환경에는 비선점형과 선점형이 있다. CAN 통신 같은 경우 하나의 메시지가 고유 우선순위를 가지는 경우 비선점형이지만, 본 논문의 IEEE1212 기반 RTU 응용계층에서 보면 여러 개의 연속적인 메시지가 고유 우선순위를 가지기 때문에 선점형 환경이다. 그렇기 때문에 고정 우선순위 선점형 스케줄링 기법에 적합한 DM(Deadline Monotonic)을 사용하여 최적화 한다.

### 2.3 시뮬레이션

앞서 설정한 우선순위 기반으로 통신 신뢰성 성능을 평가하기 위해 시간성 분석 툴 MAST(Modeling and Analysis Suite for Real-Time Applications)를 사용하였다. 시험 환경은 종류 별로 각 모듈 하나씩 총 9개가 1Mbps 네트워크에 구성되도록 하였다. 데이터는 I/O는 16개의 채널, INF는 200포인트를 주기적으로 발생하도록 하였고 UP는 각 모듈들이 대역폭을 10%씩 점유하도록 설정하였다. 그림 2는 DM 스케줄링 기법에 따라 우선순위를 부여한 MAST 화면을 보여준다. 그림 3은 계산된 WCRT와 UP와의 비를 나타낸다. 1이 넘을 경우 WCRT가 UP보다 크기 때문에 통신 신뢰성을 보장 할 수 없다. 시뮬레이션에서는 우선순위를 최적화하였기 때문에, 모든 모듈들이 통신신뢰성을 만족하였다.



<그림 2> RTU 메시지 우선순위 선정화면



<그림 3> 우선순위에 따른 WCRT vs UP 비

## 3. 결 론

CAN 통신 네트워크는 우선순위가 높은 메시지도 전송중인 메시지를 중단할 수 없는 비선점형 네트워크이다. 하지만, 응용 계층에서 의미를 가지는 연속적인 CAN 메시지가 같은 주기와 같은 우선순위를 가지고 CSMA/CR에 참여한다면, 선점형 네트워크로 간주하고 분석할 수 있다. 본 논문에서는 IEEE1212 기반의 응용계층을 가지는 RTU 네트워크에서 CAN 메시지의 ID 특성과 전송시간을 고찰하였고, 선점형 네트워크 우선순위 최적화 기법으로 알려진 DM을 이용하여, RTU 환경에서 발생할 수 있는 데이터를 설정하여 시뮬레이션을 통한 통신 신뢰성 검증을 하였다. 이러한 검증 방법은 RTU 네트워크를 설계하고 통신 신뢰성을 검증하는데 사용될 것이다.

향후 측정 혹은 제어 데이터 외에 RTU 환경에서 발생할 수 있는 메시지 혹은 변경이 불가능한 우선순위 등을 함께 분석하고, 오류로 인한 재전송을 고려하여 좀 더 실제 환경에 적합한 분석을 진행할 예정이다.

### [참 고 문 헌]

[1] K. TINDELL, A. BURNS, A.WELLINGS, "Calculating Controller Area Network(CAN) Message Response Times", Control Engineering Practice, vol.3, no.8 , 1163-1169, 1195  
 [2] N. Audsley, A. Burns, M. Richardson, K. Tindell, A.J. Wellings, "Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling", Software Engineering Journal, vol.8, no.5, 284-292, 1993  
 [3] Robert I. Davis, Alan Burns, Reinder J. Bril, Johan J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised", Real-Time Systems, vol.35. no.3, 239-272, 2007