

DDS 기반 자율 주행 시스템의 시간적 제약 위반 탐지 기법

안재호^o, 노순현^{*}

^o서울대학교 전기정보공학부

e-mail: {jhahn, shnoh}@redwood.snu.ac.kr^{o*}

Detection Mechanisms for Timing Constraint Violations in DDS-Based Autonomous Driving System

Jae-ho Ahn^o, Soon-hyun Noh^{*}

^{o*}Department of Electrical and Computer Engineering, SNU

● 요약 ●

자율 주행 자동차는 다수의 센서와 ECU 등으로 구성된 분산 시스템이다. 이 시스템은 다양한 시간적 제약사항들을 갖는 자율주행 응용들을 구동하며 각 응용들에 대한 시간적 제약사항 위반을 탐지해야한다. 이러한 분산 시스템에서 응용들 간의 통신을 위해 사용되는 미들웨어들 중 대표적인 것은 DDS이다. DDS는 높은 확장성을 지원하는 발행-구독 통신 모델을 기반으로 하며, 실시간성을 고려한 다양한 QoS 정책들을 제공한다. 하지만 DDS는 자율주행 응용이 요구하는 시간적 제약사항들 중 deadline과 correlation 제약 사항에 대한 위반 여부를 탐지하지 못한다. 본 논문은 DDS 기반 시스템에서 deadline과 correlation 제약 사항 위반 여부를 런타임에서 탐지하는 기법을 제안한다. 본 연구진은 제안된 기법을 DDS의 구현들 중 하나인 Vortex 사의 OpenSplice 기반 시스템에 구현하였다. 실험을 통해 검증한 결과, deadline과 correlation 제약 사항에 대한 위반 여부를 적은 오버헤드와 함께 성공적으로 탐지하였다.

키워드: Timing constraints, violation detection, DDS (Data Distribution Service), Splash

1. 서론

자율 주행 자동차에는 운전자의 안전을 위한 주요 기능들로 ADAS (Advanced Driver Assistance System)나 PCS (Pre-crash Safety System)같은 기능이 탑재 된다. 이 기능들이 실시간으로 동작하기 위해서는 차량 내부 및 외부 상황 정보를 송수신함에 있어서 실시간 통신이 필수적이다. 최근 들어 자율주행 기능이 더욱 고도화됨에 따라 자율주행 자동차에 요구되는 ECU, 센서, 응용의 개수가 기하급수적으로 증가하고 있다[1]. 이러한 상황에서 높은 확장성과 실시간성 관련 기능을 제공하는 통신 미들웨어가 매우 중요해지고 있다. 다양한 통신 미들웨어들 중 DDS (Data Distribution Service)는 실시간성을 위한 QoS 보장 기법을 제공하며, 자율 주행 자동차 같은 분산 시스템에 적합한 높은 확장성을 가지는 publish-subscribe 통신 모델에 기반을 두어 동작한다. 이로 인해 자율 주행을 위한 다양한 응용 프로그램에 DDS 사용이 많이 시도 되고 있다[2]. 구체적으로, 개방형 차량 소프트웨어 아키텍처 표준인 AUTOSAR가 자율 주행 자동차를 위해 발표한 adaptive platform에서는 DDS를 고려한 새로운 표준이 발표 되었다 [3]. 또한 많은 개발자들은 실시간성 보장 기법을 도입한 automotive Ethernet 표준인 TSN (time sensitive network)에도 DDS를 탑재하기 위해 개발 중이다[5].

하지만 DDS는 자율 주행 응용이 요구하는 다양한 시간적 제약사항

들 중 일부를 고려하지 않고 있다. 우리는 자율 주행을 위한 스트림 프로세싱 언어인 Splash를 개발하며 자율 주행에서 요구되는 시간적 제약 사항을 도출하였다[6]. 본 연구진은 Splash에서 도출된 시간적 제약 사항과 DDS의 기능을 비교 분석한 결과, DDS는 deadline과 correlation 제약 사항들의 위반 여부를 탐지하지 못한다는 것을 확인하였다. 우리는 이러한 한계를 극복하기 위해 DDS 기반 시스템의 deadline과 correlation 제약 사항 위반탐지 기법을 제시한다. 이 기법을 DDS의 오픈소스 구현인 OpenSplice DDS 상에서 실험을 하였으며 시간적 제약 사항 위반 여부를 런타임에서 적은 오버 헤드로 탐지하였다.

본 논문의 나머지 장은 다음과 같이 구성되어 있다. 2장에서는 논문의 이해를 돕기 위한 DDS의 기본 구조와 원리를 설명한다. 3장에서는 DDS가 지원하지 못하는 자율 주행에서 요구되는 시간적 제약 사항에 대해 설명을 한다. 4장에서는 DDS가 지원하지 못하는 시간적 제약 사항에 대한 위반 탐지 기법을 자세하게 설명한다. 5장에서는 OpenSplice DDS로 구현된 응용 프로그램을 사용하여 탐지 기법에 대한 실험 및 검증을 설명한다. 마지막으로 6장에서 개발된 탐지 기법에 대한 결론을 다룬다.

Table 1. 시간적 제약 사항을 위한 DDS의 지원

시간적 제약 사항	설명	제약 사항의 위반 탐지를 위한 DDS 지원
Input/output rate 제약 사항	단위 시간 당 입력 받아야 하는 데이터와 출력해야 하는 데이터의 최소 개수이다.	Deadline QoS 정책을 사용하여 탐지가 가능하다. Deadline QoS 정책은 DataWriter의 출력과 DataReader의 입력 사이의 최대 허용 시간을 설정해준다.
Freshness 제약 사항	데이터의 출력 시간부터 처리 될 수 있는 유효 기간이다.	Lifespan QoS 정책을 사용하여 탐지가 가능하다. Lifespan QoS 정책은 데이터가 DataWriter에서 출력되는 순간부터의 유효 기간을 설정해 준다.
Deadline	입력되는 데이터가 처리되어 최종 결과가 출력되는 시간까지의 제한 시간이다.	지원되지 않는다.
Correlation 제약 사항	함께 처리 돼야 할 데이터들 간의 생성 시간 차이이다.	지원되지 않는다.

II. DDS의 구조와 원리

DDS는 publish-subscribe 통신 모델을 사용한다. publish-subscribe 통신 모델이란 publisher가 named channel인 topic에게 데이터를 송신하면 subscriber는 관심 있는 topic으로부터 원하는 데이터만 수신하는 통신 모델이다. 송수신 시 publisher와 subscriber는 서로의 존재를 몰라도 되며 송신할 topic과 수신할 topic만 알면 통신을 할 수 있다.

DDS는 DCPS(data centric publish and subscribe) 계층과 RTPS(real-time publish and subscribe) 계층으로 구성되어있다. 상위 계층인 DCPS는 응용 프로그램에서 DDS의 기능을 사용하기 위해 통신 및 QoS 설정 인터페이스를 제공하며 하위 계층인 RTPS는 publish-subscribe 통신을 기반으로 응용 프로그램 간의 데이터 전송, 발견, 그리고 상호 운용 성을 제공한다[4]. 본 논문은 QoS 관리가 이뤄지는 DCPS layer를 중심으로 연구를 진행하였다.

DDS 응용 프로그램에서는 publish-subscribe 통신 모델의 publisher 역할을 하는 DataWriter를 통해 생성된 데이터를 하나의 topic으로 출력한다. 그 후 별도의 응용 프로그램에서는 subscriber 역할을 하는 DataReader를 통해 하나의 topic에서 데이터를 입력받는다. 응용 프로그램은 여러 개의 DataReader와 DataWriter를 가질 수 있다. DataReader와 DataWriter는 객체로 제공이 되며 객체의 멤버 함수를 통해 입출력을 할 수 있다. 따라서 DDS 응용프로그램이 여러 쓰레드로 구성될 경우 각 쓰레드는 DataReader와 DataWriter의 객체를 접근할 수 있다.

또한 DCPS layer는 분산 실시간 통신을 위한 22가지 QoS 보장 정책을 지원한다[4]. 각 QoS 정책은 실시간 통신에서 지켜져야 할 제약 사항을 사용자가 설정할 수 있도록 해준다. 설정된 QoS 정책은 실시간으로 모니터링 하여 제약 사항이 지켜지지 않은 경우 개별적인 쓰레드로 구현된 핸들러를 호출한다.

III. 문제 정의

우리는 Splash를 통해 자율 주행에서 요구되는 시간적 제약 사항들을 도출하였다. 도출된 시간적 제약 사항들은 총 4가지이며 input/output rate 제약 사항, freshness 제약 사항, deadline, correlation 제약 사항으로 구성 되어 있다[6]. 표1에서는 시간적 제약 사항에 대한 설명과 그와 관련된 DDS 지원 내용이 설명되어있다. 본 논문에서는 도출 된 시간적 제약 사항들에 대한 위반이 DDS의

기능으로 탐지가 가능한지를 DDS의 QoS 정책에 대한 분석을 통해 확인하였다. 그 결과 input/output rate 제약 사항과 freshness 제약 사항은 DDS의 deadline QoS 정책과 lifespan QoS 정책으로 탐지할 수 있음을 확인하였다. 하지만 현재 DDS의 기능으로는 deadline과 correlation 제약 사항에 대한 위반 탐지 기능이 없다는 것을 확인하였다. 이에 따라 본 논문에서는 DDS를 다음과 같이 보완한다.

(1) Deadline 제약 사항에 대한 명시와 위반에 대한 탐지 기능을 개발한다.

(2) Freshness 제약 사항에 대한 명시와 위반에 대한 탐지 기능을 개발한다.

IV. Timing 제약 사항 탐지 기법

4장에서는 deadline과 correlation 제약 사항에 대한 위반 탐지 기법을 상세하게 설명한다. 4.1절과 4.2절에서는 각 deadline 그리고 correlation 제약 사항 위반 탐지 기법의 구조, 핵심 원리, 그리고 세부 사항을 설명한다.

4.1 Deadline 위반 탐지 기법

Deadline의 위반 탐지 기법을 구현하기 위해서는 DataWriter의 데이터 송신 핸들러와 DataReader의 데이터 수신 핸들러가 수정되었다. Deadline 제약 사항이 적용된 DataWriter와 DataReader는 각각 타임스탬프를 관리하기 위한 FIFO 큐를 가진다. FIFO 정책으로 운영되는 queue를 가지는 이유는 가장 최근 출력으로 이어지는 입력 data의 time stamp를 사용하여 제약 사항 위반을 탐지하기 위해서다.

Deadline은 입력되는 데이터가 처리되어 최종 결과가 출력되는 시간까지의 제한 시간이다. Deadline 위반 탐지 기법은 다음과 같다. DataReader는 데이터가 입력되는 순간 데이터의 입력 시간을 자신의 FIFO 큐에 타임스탬프로 보관한다. DataWriter는 데이터가 출력되는 순간 데이터의 출력 시간을 자신의 FIFO 큐에 타임스탬프로 보관한다. 각 DataReader와 DataWriter는 가장 최신 출력으로 이어진 타임스탬프를 사용하기 위해 자신의 FIFO 큐의 head로부터 타임스탬프를 가져와 둘의 차이 값을 구한다. 설정된 deadline보다 차이 값이 더 큰지 확인하여 위반을 탐지한다. 위반 탐지 기법은 deadline이 적용된 DataWriter가 출력을 할 때 마다 이루어진다.

Deadline의 탐지는 한 개 이상의 입력이 하나의 출력으로 이어질 경우 어느 입력의 타임스탬프가 출력의 타임스탬프와 비교 될지

정해 저야 한다. 따라서 본 논문에서는 출력 타임스탬프와의 차이가 가장 많이 나는 입력 타임스탬프를 사용한다. 이를 달성하기 위해서 deadline이 적용되는 모든 DataReader들은 DataWriter가 출력을 하는 순간, 각 FIFO 큐의 head를 비교하여 가장 입력 시간이 빠른 타임스탬프를 선정한다. 선정된 타임스탬프는 DataWriter의 FIFO 큐 head로부터 뺀 값을 deadline과 비교해서 위반을 탐지 한다.

한 DataWriter에게는 한 개의 deadline만 설정될 수 있지만 한 DataReader의 입력은 여러 DataWriter의 출력으로 이어 질 수 있다. 그러므로 한 DataReader에게는 여러 deadline이 적용될 수 있으며 적용되는 각 deadline의 개수만큼 FIFO 큐를 가진다.

4.2 Correlation 제약 사항 위반 탐지 기법

Correlation 제약 사항의 위반 탐지 기법을 구현하기 위해서 DataReader의 데이터 수신 핸들러가 수정되었다. Correlation 제약 사항이 적용되는 각 DataReader는 타임스탬프를 관리하기 위해 FIFO 큐를 생성한다. FIFO 정책으로 운영되는 queue를 가지는 이유는 DataReader들 간의 가장 최근 입력된 time stamp를 사용하여 제약 사항 위반을 탐지하기 위해서다.

Correlation 제약 사항은 2개 이상의 입력으로부터 들어오는 데이터들 간의 생성 시간을 비교한다. 그 후 비교를 통해 구해진 생성 시간 간의 차이 값을 제약한다. 여기서 생성 시간이란 DataReader로 입력되는 데이터가 입력되기 전 DataWriter로부터 출력되는 시간을 뜻한다. DDS는 DataWriter가 출력하는 raw 데이터 외에도 부수적인 메타데이터를 출력 데이터에 포함 시켜준다. 이 메타데이터에는 생성 시간이 DataWriter가 데이터를 출력 하는 시간으로 포함이 되어 있다. DataReader는 데이터가 입력되는 순간 생성 시간을 접근 할 수 있다. 위에 설명된 기능을 사용하여 correlation 제약 사항 위반을 탐지 하는 기법은 다음과 같다. Correlation 제약 사항의 탐지 대상인 DataReader들은 각 자신의 데이터가 입력되는 순간 데이터의 생성 시간 타임스탬프를 자신의 FIFO 큐에 넣어준다. 탐지 대상인 DataReader들은 가장 먼저 들어온 타임스탬프들을 비교하기 위해 FIFO 큐의 head로부터 타임스탬프를 가져온다. 가져 온 타임스탬프 중 가장 최신 시간을 가진 타임스탬프와 가장 오래된 타임스탬프를 비교하여 차이 값을 구한다. 구해진 차이 값이 correlation 제약 사항보다 더 크지 확인하여 위반을 탐지한다. Correlation 제약 사항은 2개 이상의 DataReader에게 적용되는 제약 사항이므로 어느 시점에서 생성 시간 간의 차이 값이 계산되는지 정해야 한다. 그러므로 본 논문에서는 correlation 제약 사항이 적용되는 모든 DataReader의 FIFO 큐에 생성 타임스탬프가 입력되는 순간 탐지 기법이 시작한다.

한 DataReader는 여러 correlation 제약 사항의 대상이 될 수 있다. 이런 경우 적용되는 각 correlation 제약 사항 개수만큼 FIFO 큐를 가진다.

V. 실험 환경 및 검증 결과

위의 기법은 DDS vendor사인 Vortex의 OpenSplice DDS로 실험하였다. 실험을 위한 응용 프로그램은 3개의 DataReader

R_1, R_2, R_3 와 2개의 DataWriter W_1 과 W_2 로 입출력을 한다. 응용프로그램 예제는 두 개의 deadline과 하나의 correlation 제약 사항이 적용된다. 첫 번째 deadline은 30ms이며 R_1, R_2, W_1 에게 적용된다. 두 번째 deadline은 50ms이며 R_2, R_3, W_2 에게 적용된다. Correlation 제약 사항은 10ms이며 R_1, R_2, R_3 에게 적용된다. 응용 프로그램은 2개의 쓰레드 T_1 과 T_2 로 구성 되어있다. T_1 은 입출력을 제외한 10ms의 처리 시간을 가지며 R_1 과 R_2 로 입력을 받고 W_1 을 통해 출력한다. 첫 번째 deadline은 T_1 의 처리 시간을 감사한다. T_2 은 입출력을 제외한 20ms의 처리시간을 가지며 R_2 와 R_3 로 입력을 받고 W_2 로 출력한다. 두 번째 deadline은 T_2 의 처리 시간을 감지해 준다. R_1, R_2, R_3 은 각 2ms, 4ms, 6ms의 주기로 데이터가 생성 되어 입력된다. Correlation 제약 사항은 이 세 개의 DataReader로 입력되는 데이터의 생성 시간차이를 감시한다.

응용 프로그램의 초기 상태는 제약 사항의 위반 없이 실행이 된다. 기법 검증을 위해 인위적인 지연 상황을 발생 시킨다. DataReader들에게 입력될 각 데이터의 생성 주기를 각 2ms, 8ms, 16ms로 늦추어 correlation 제약 사항 이상으로 변경되게 하였으며 각 쓰레드 T_1 와 T_2 의 처리 시간도 50ms와 70ms의 처리 시간으로 변경이 되게 하여 설정된 deadline 이상의 latency를 준다. 각 제약 사항의 위반 시 핸들러를 호출하도록 하여 위반 상황을 확인하였다.

본 실험에서는 두 쓰레드 T_1 과 T_2 에서 발생한 지연으로 인한 deadline 위반을 성공적으로 탐지 하였으며 늦춰진 생성 시간으로 인해 발생하는 correlation 제약 사항의 위반을 성공적으로 탐지 할 수 있었다.

Table 2, 실험 환경

하드웨어	
CPU	Intel Core i7-7700 3.6G Hz
메모리	DDR4 8GB
스토리지	25.6GB
소프트웨어	
DDS	OpenSplice DDS 6.7.1
운영 체제	Ubuntu 16.04 LTS

VI. 결론

본 논문에서는 DDS에서 지원하지 못하는 자율 주행 응용 프로그램에서 요구되는 시간적 제약 사항 위반에 대한 탐지 기능을 제안하였다. 본 연구는 OpenSplice DDS 버전 6.7.1에 제안된 기법을 적용하였고, 표 2의 실험 환경에서 실행한 검증을 통해 제약 사항에 대한 위반을 성공적으로 탐지하였다.

ACKNOWLEDGEMENT

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신

신기술진흥센터의 지원을 받아 수행된 연구임 (No R7117-19-0164,
차량 통신 기반의 광역 주행환경인지 및 협조 주행기술 개발)

REFERENCES

- [1] Huang, Jun, et al. "In-Vehicle Networking: Protocols, Challenges, and Solutions." IEEE Network (2018).
- [2] Abdellaoui, Zouhaira, and Majdi Benzarti. "Real time communication on DDS over FlexRay using SAE benchmark model." Control, Automation and Diagnosis (ICCAD), 2017 International Conference on. IEEE, 2017.
- [3] Adaptive Adaptive Platform 18.03, AUTOSAR SWS_C ommunication Management (Section 7.3.2)
- [4] Pardo-Castellote, Gerardo, Bert Farabaugh, and Rick Warren. "An introduction to DDS and data-centric communications." RTI, Aug (2005).
- [5] Kugele, Stefan, David Hettler, and Jan Peter. "Data-Centric Communication and Containerization for Future Automotive Software Architectures."
- [6] Soonhyun Noh and Seongsoo Hong, "Splash: Stream Processing Language for Autonomous Driving," Real-Time and Embedded AI for Autonomous Vehicles (WREAV), Jun 2018