논문 16-41-12-24 The Journal of Korean Institute of Communications and Information Sciences '16-12 Vol.41 No.12
https://doi.org/10.7840/kics.2016.41.12.1814

국방 훈련체계용 가상물리시스템 시간 동기화 기법

홍 석 준◆, 이 우 엽*, 조 인 휘*, 김 원 태°

Time Synchronization Scheme of Cyber-Physical Systems for Military Training Systems

Seok-Joon Hong◆, Woo-Yeob Lee*, In-Whee Joe*, Won-Tae Kim°

요 약

LVC(Live Virtual Constructive) 통합 훈련 체계는 대표적인 가상물리시스템 중 하나이다. LVC 훈련 체계를 구성하는 각 L,V,C 체계들은 서로 다른 시간 도메인, 해상도 및 동작 방식을 갖는다. 이에 각 체계의 이종 미들웨어들을 공통의 통신 미들웨어로서 통합하고, 연동 gateway (GW)를 이용하여 이종 체계를 연동하는 방안 연구는 매우 중요하다. 특히, LVC체계는 서로 다른 시간 도메인을 사용하고 있기 때문에 다른 체계 간 이벤트들의 인과성 보장과 시간 동기화에 대한 연구가 필요하다. 본 연구에서는 LVC 훈련체계의 통합 미들웨어로서 L 체계의 실시간 통신 미들웨어인 OMG DDS(Data Distribution Service)를 기반으로 HLA(High Level Architecture)/RTI(Run Time Infrastructure)기반의 시뮬레이션 시간을 사용하는 V, C 체계와 시간 차원에서 연동하는 방안을 제시한다. 본 논문에서는 HLA의 Time management 기능과 Clock Federate를 기반으로 DDS와 HLA 각각의 통신 개체인 파티서펀트(Participant)와 페더레이트(Federate)간 인과성을 유지하는 정밀한 시간 동기화 기법을 제안한다. 더불어, 각 미들웨어 상에서 실제 응용을 개발하여 제안한 방식에 의해 이종 체계 간 시간 제어 및 동기화가 이루어짐을 검증한다.

Key Words : CPS, LVC middleware, DDS, HLA/RTI, Time management

ABSTRACT

LVC(live-virtual-constructive) integrated training system is a representative cyber-physical system. Each systems in a LVC system has different time domain, resolution and operation methods. So, it is very important to integrate different middlewares as a common middleware for heterogeneous systems using inter-working GWs. Especially, since the LVC system uses different time, it is necessary to study the method for guaranteeing causality and time synchronization among the events from different systems. In this study, we propose an time synchronization scheme to integrate the virtual and constructive system which use the simulation time of HLA (High Level Architecture)/ RTI (Run Time Infrastructure) into the live system based on the OMG DDS (Data Distribution Service). We propose a precise time synchronization scheme based on HLA time management and clock federate between participants and federates which are the communication objects of DDS and HLA/RTI respectively. In addition, we verified that time is well-synchronized among heterogeneous systems using the suggested scheme by implementing and demonstrating simulation applications on each middleware.

※ 본 연구는 미래창조과학부 정보통신기술센터 SW컴퓨팅산업원천기술개발사업의 연구비지원(R0114-16-0046)에 의해 수행되었습니다.
◆ First Author : Department of Electronics and Computer Engineering, Hanyang University, daniel379@hanyang.ac.kr, 학생회원
° Corresponding Author : Korea University of Technology and Education, wtkim@koreatech.ac.kr, 정회원
* Department of Electronics and Computer Engineering, Hanyang University, {matias12, iwjoe}@hanyang.ac.kr
논문번호 : KICS2016-09-276, Received September 28, 2016; Revised November 2, 2016; Accepted November 14, 2016

Ⅰ. 서 론

1.1 연구의 배경

현대의 육군은 과학화 훈련 체계 발전을 위해 노력하고 있다. 그중에서도 컴퓨터 시뮬레이션[1]이나 가상물리 시스템(CPS: cyber-physical system)[2]과 같은 첨단 과학 기술을 통한 국방 훈련은 실제 훈련에 소모되는 경비와 시간, 장소 제약 등의 문제를 해결하는 등 국방 훈련의 과학화에 큰 기여를 하고 있다. LVC 통합 훈련은 가상물리시스템을 이용한 과학화 훈련체계 중 하나로, 독립적으로 운용되는 서로 다른 실 기동 모의 체계(Live), 가상 모의 체계 (Virtual) 구성 모의 체계(Constructive)를 상호 연동하여 공통 작전 상황도(COP: common operational picture)에 통합 시현함으로써 기존의 독립적인 운용보다 상하 제대 간 동시성, 통합성, 실전성 등을 향상시킨 과학화 훈련 방법이다[3].

L, V, C 각각의 시스템들은 각자 다른 운영 체계와 프로토콜을 사용하여 동작이 되기 때문에 LVC 연동을 위해서는 기존의 서로 다른 시스템을 연동하기 위한 방법 및 LVC 통합 통신 미들웨어에 대한 연구와 개발이 필요하다[4-7].

최근에는 서로 다른 체계 기반의 L, V, C 시스템을 연동하는 통신 미들웨어가 연구 개발된 사례가 있다. 이 연동 미들웨어에서 각 체계를 대표하는 미들웨어로는 L 체계는 DDS[8-9]를 V와 C 체계를 대표하는 미들웨어로는 IEEE HLA/RTI [10-13]가 사용되었다.

LVC 미들웨어의 중요한 기능 중의 하나는 서로 다른 시스템간의 시간을 동기화하는 것이다. 즉, DDS 기반의 Live 시스템의 시간은 실시간으로 동작하고, HLA기반의 V와 C체계는 논리시간 즉 시뮬레이션 시간을 사용하기 때문에 서로 다른 체계가 같은 시간 진행 속도를 가질 수 있도록 시간을 동기화하는 것은 매우 중요하다. 예를 들어, 통합 훈련 시나리오가 Live 요소인 병사들과 Virtual 요소인 탱크가 있고, 병력과 탱크의 이동을 같이 볼 수 있는 Constructive 요소인 전투지휘센터에서 관찰하는 경우, 실제 요소와 가상 요소가 동일 시점에서 이동을 시작하여 동일한 속도로 목표 지점까지 이동하는 것과 같은 실시간 연동 시나리오는 시간 동기화가 이루어지지 않는다면 각 체계에서의 일어나는 이벤트들의 인과관계가 손상되어 시뮬레이션의 결과는 믿을 수 없게 된다.

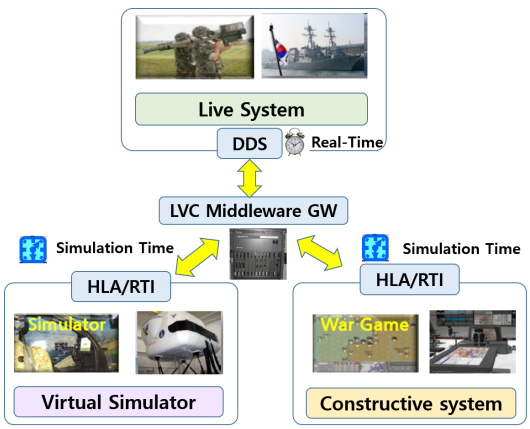

그림 1. LVC 통신 미들웨어 GW를 통한 LVC 연동
Fig. 1. LVC inter-working through LVC middleware GW

1.2 관련 연구

LVC 시스템 연동에 대한 기존의 연구들에서는 시간 동기화 즉 Live 시스템의 실시간에 맞춰 시간을 진행하기 위해서 V, C 시스템들이 실시간으로 동작하기 위한 기법들을 제시하였다. 어떤 연구에서는 V, C 시스템에서 생성되는 데이터를 실시간으로 처리하기 위한 기법을 제시하였고[4], 또 다른 연구들은 V, C 시스템에서 GPS를 사용하여 실시간 위치 정보를 가지고 시뮬레이션 하는 방법을 제안하기도 하였다[5].

하지만, 기존 연구들에서는 LVC간 연동에 있어서 별도의 시간 관리 기능을 사용하여 시간을 동기화하지 않았기 때문에, 각 시스템에서 발생한 이벤트 정보를 담은 메시지들이 네트워크 지연과 같은 문제가 발생하게 될 경우, 시간에 따른 인과성이 깨질 수 있는 여지가 있었다.

따라서 본 논문에서는 LVC 미들웨어에서 DDS의 실시간 진행 속도에 HLA/RTI 기반의 시뮬레이션 시간을 동기화하는 알고리즘을 제안한다. 이 알고리즘은 시뮬레이션 객체들(프로그램)의 시뮬레이션 시간의 진행 속도를 조절하기 위해서 HLA/RTI에서 제공하는 Time management 기능과 실시간에 맞춰 시간을 진행하고 Federation내 다른 Federate들의 시간을 조절하는 역할을 하는 Clock Federate를 사용하여 실시간에 시뮬레이션 시간을 동기화한다. 또한 제안한 기법을 적용한 테스트 수행하였으며, 수행한 시뮬레이션의 결과는 Time management와 Clock Federate를 사용했을 때, DDS의 실시간 진행에 맞춰 HLA/RTI기반의 시뮬레이션 Federate들의 시뮬레이션 시간들이 동기화될 수 있음을 보여준다.

본 논문은 다음과 같이 구성된다. Ⅱ장에서는 HLA Federation에서 어떻게 Time management를 통해 시간 동기화가 이루어질 수 있는지를 살펴보고, Ⅲ장에

서는 이를 LVC GW에 적용하는 방법을 소개한다. Ⅳ과 Ⅴ장에서는 제안한 알고리즘을 통해 어떻게 시간 동기화를 이룰 수 있는지 LVC 시연 연동 시스템과 테스트 시나리오를 통해 확인하고, Ⅵ장에서 결론을 맺는다.

Ⅱ. HLA Federation의 실시간 동기화

HLA/RTI는 연동 가능한 분산 시뮬레이션을 위해 그림 2와 같이 RTI를 기반으로 하나의 Federation을 구축하며, 이를 통해 여러 개의 시뮬레이션 프로그램(Federate)들이 RTI의 Object management 서비스를 이용하여 Federation내의 다른 Federate들에게 Update Attribute Values(UAV) 혹은 Send interaction을 사용하여 데이터를 전송하거나 Reflect Attribute Values(RAV) 혹은 Receive interaction을 사용하여 수신할 수 있다[11].

또한, RTI의 Time management 서비스를 이용하면 각 Federate들이 각자 자신의 시뮬레이션 시간을 갖고 시뮬레이션에 참여할 수 있는데, 이때 각 Federate들은 오직 RTI를 통해서 다음 시뮬레이션 시간을 요청(TAR: Time Advance Request)하고, 요청한 시간에 대한 허락(TAG: Time Advance Grant)를 받았을 경우에만 다음 시뮬레이션 시간으로 진행할 수 있다[11].

그림 3은 같은 시간 간격으로 진행하는 하나의 Time stepped Federate가 한 시뮬레이션 주기 동안 어떻게 RTI를 통해서 데이터를 주고 받으며 시뮬레이션 시간을 진행시키는지 보여준다. 여기서 wallclock time이란 현재 시점의 실제 시간(Real time)을 말한다[14,15].

또한, 위 그림 3에서 Federate의 Logical time과 wallclock time의 관계는 다음 공식 (1)과 같이 나타낼 수 있다. 여기서 만약 Scale 값이 2 이라면 실제 시간보다 2배 빠른 시뮬레이션이고, Scale값이 1이라면, 실시간에 동기를 맞춘 시뮬레이션이 된다.

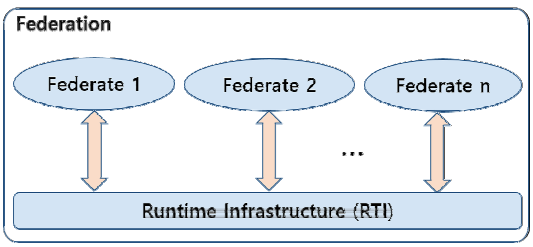

그림 2. HLA/RTI를 통한 분산 시뮬레이션
Fig. 2. Distributed simulation through HLA/RTI

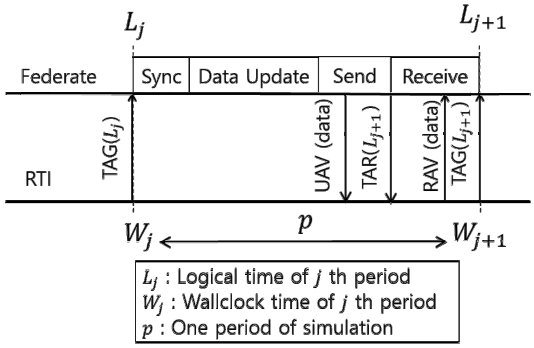

그림 3. Time stepped Federate의 한 주기 동안의 동작
Fig. 3. Operation during one period of time stepped federate

$$L_{j+1} - L_j = Scale * (W_{j+1} - W_j) \quad (1)$$

그림 4에서는 하나의 Federation내에 2개의 Time stepped Federate (Fed1, Fed2)가 다른 time step (Fed1 : 1, Fed2 : 2)을 가지고 같은 시간에 시간 진행을 요청(TAR)할 때, RTI가 어떻게 시간 진행을 허락(TAG)해주는지를 보여준다.

즉, 시뮬레이션이 수행되면서 RTI가 같은 시간 ($t_0 + 1$)에 Fed1과 Fed2로부터 시뮬레이션 시간을 각각 t+1와 t+2 으로 진행 요청을 받게 되면 RTI는 먼저 Fed1의 t+1로 시간 진행을 허락해준 후 다음에 다시 Fed1이 t+2를 요청했을 때, Fed1과 Fed2의 시간을 같이 t+2로 진행시켜준다. 이런 식으로 RTI는 Federation내 모든 Federate들의 TAR을 받아서 진행할 수 있는 최소 진행 가능한 시뮬레이션 시간부터 시뮬레이션 시간을 진행시켜준다.

HLA/RTI에서 제공하는 메시지 순서는 RO

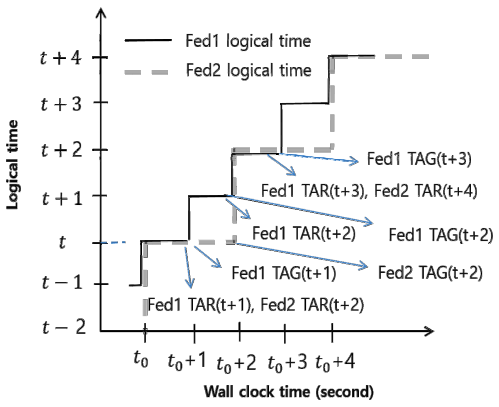

그림 4. 두 개의 Time stepped Federate의 시간진행
Fig. 4. Time advance of two time stepped federates.

(Receive Order)와 TSO(Time stamp order)이 있다. RO는 메시지가 수신되는 순서로 이벤트를 정렬하는 방식이다. 반면에 TSO (Time stamp order)는 메시지에 있는 time stamp값을 사용하여 정렬하는 방식으로서 송신자는 항상 메시지에 time stamp를 넣어 전송하고, RTI는 이를 순서 정렬하여 수신측에 전송해주는 방식이다[11].

다음은 Time management를 설정하는 두 가지 옵션에 대한 설명이다.

- Time regulating : 이 설정이 Enable되어 있으면 Federate는 자신의 Logical time을 이용하여 TSO 메시지를 전송할 수 있다. 이후 RTI는 이를 정렬하여 다른 Federate로 전송해준다.
- Time constrained : 이 설정이 Enable되어 있으면 RTI를 통해서 TSO 메시지를 시간 순서대로 수신할 수 있다. 하지만 Disable되어 있으면 Federate는 모든 메시지를 RO 메시지(Time stamp가 없는)로 수신하게 된다.

다음은 위의 Time management 설정에 따른 4가지 종류의 Federate를 나타낸다[16].

- Logical time synchronized : 이 타입의 Federate는 Time regulating이면서 Time constrained인 경우이다. 이 경우 다른 Federate들의 시간 진행에 영향을 미치기도 하고 또한 영향을 받기도 한다.
- Externally time synchronized : 이 Federate는 Time regulating과 Time constrained 둘 다 아닌 경우이다. 그것은 자신의 내부의 시간을 RTI가 제공하는 것이 아닌 다른 메커니즘을 이용하여 시간을 진행한다. DIS(Distributed Interactive Simulation)의 경우가 대표적인 예로서 wallclock time이 동기화에 사용된다.
- Logical time passive : 이 경우 Time constrained만 설정된 경우로서 Federate의 시간은 다른 Federate들의 영향을 받아서 진행하고 영향을 주지는 않는다. 관찰자 Federate나 Federation 관리 툴이 그런 예가 될 수 있다.
- Logical time aggressive : 이 경우 Time regulating만 설정된 경우로서 Federate는 다른 Federate들의 시간 진행에 영향을 미치지만, 자신의 시간 진행에 있어서는 다른 Federate들의 영향을 받지 않는다. 이 종류의 Federate를 사용하여 Federation의 전체 시간 진행을 이 Federate의 시간 진행 페이스에 맞춰 진행하게 할 수 있다.

HLA/RTI 개발자들은 이런 Time management의 기능을 이용하여 실시간에 동기를 맞춰 진행하는 Federation을 구성하는 방법을 소개하였다[16]. 즉, 하나의 Federate를 Time regulating만 설정하고, 실시간과 동기를 맞춰서 진행하도록 한 후에, Federation내의 동기화가 필요한 다른 Federate는 Time constrained only혹은 Time regulating and constrained로 설정하면 RTI의 시간 관리를 통해 Federation이 실시간과 동기를 맞춰 진행할 수 있게 된다. 여기서 실시간과 동기를 맞추면서 다른 Federate들의 시간을 조절하는 역할을 하는 Federate를 “Clock Federate“라고 부른다[14].

Ⅲ. LVC GW 기반 시간 동기화 기법

그림 5는 LVC 연동을 위한 GW의 구조를 보여주고 있다. 현재의 연구는 LVC 연동에서 DDS 파트에서 HLA 파트로 데이터를 전송하는 경우에 대한 인과성 보장과 시간 동기화에 초점을 맞추었다.

GW는 양쪽의 다른 미들웨어들과의 연동을 위해서 두 개의 미들웨어를 모두 지원하며 DDS 파트로부터 데이터가 수신 때마다 DDS의 Participant와 HLA/RTI의 Federate를 맵핑하여 데이터를 전달한다. 예를 들면 DDS domain에서 하나의 participant가 data를 publish하면 LVC GW는 자신의 DDS participant를 통해 데이터를 수신하고 이를 자신의 Federate를 통해 송신하여 최종적으로 HLA domain의 Federate들에게 전달되도록 한다. LVC GW에는 DDS의 실시간에 동기를 맞추면서 HLA domain의 Federate들의 시간을 동기화해주는 역할을 하는 Clock Federate를 별도로 실행하여 HLA 파트에 있는 Federate들과 함께 하나의 실시간 Federation으로 동작하도록 한다.

그림 6은 그림 5의 구조에 따라 실제로 구현된 GW Federate의 Pseudo 코드를 보여준다. 이 코드를 통해

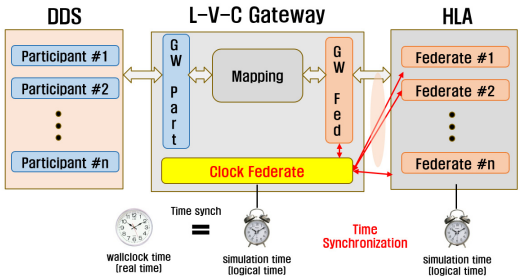

그림 5. LVC GW 구조 및 LVC 동기화 방법
Fig. 5. LVC GW architecture and LVC synchronization method

```
/* LT is logical time for this federate */
LT= 0
timeStep = 1
lookAhead = 1

while (SimulationEnd != TRUE) do
        /* Data update part  (Read DDS data and Send to HLA part)*/
        buffer.readDDSPartData(&BallData)
        timeStamp = LT
        timeStamp += lookAhead
        rtiAmb.updateAttributeValues(BallData, timeStamp)

        /* Time advance part */
        LT += timeStep
        rtiAmb.timeAdvanceRequest(LT)
        while (TimeGranted != TRUE) do
              //use "tick()" wait callback from RTI.
              rtiAmb.tick()
        end while
end while
```

그림 6. GW Federate의 Pseudo 코드
Fig. 6. Pseudo code of GW Federate

서 GW는 시뮬레이션이 끝날 때까지 매 주기마다 DDS로부터 전달된 데이터(이 코드의 경우 볼의 위치 정보)를 버퍼에서 가져와서 현재 자신의 Federate의 Logical time의 시간에 Lookahead을 더한 값을 time stamp로 설정한 후 RTI에게 UAV를 사용해서 TSO 메시지를 전송한다. 여기서 Lookahead 값은 0보다 큰 값으로서 RTI의 시간 진행을 위해서 사용되며 모든 Regulating Federate는 time stamp값을 자신의 Logical time + Lookahead 값보다 작은 값을 사용할 수 없다[11]. 여기서는 기본으로 세팅된 값인 '1'을 사용하였다. 이후에 RTI는 Federation 내에 있는 모든 Regulating Federate들 중에서 Logical time + Lookahead의 최소 값을 보고 Constrained Federate들의 시간 진행을 허락해준다. 이후 Federate는 자신의 현재 시간에서 하나의 time step만큼 증가시킨 시뮬레이션 시간을 RTI에게 TAR을 통해 요청한다. 그러면 RTI는 현재 Federation내에 있는 다른 모든 Federate의 진행 시간을 고려해서 일정 시간 이후에 TAG를 통해 시뮬레이션 시간을 진행시켜준다.

그림 7은 Clock Federate의 Pseudo 코드를 보여준다. GW Federate와 다른 점은 Clock Fedrate의 경우, 시간 진행만을 위한 Federate이기 때문에 데이터를 업데이트하는 부분이 없다는 것과 시간 동기화를 위한 코드가 포함된 것이다. 시간 동기화를 위해서 Clock Federate는 자신의 Logical time의 진행 속도를 wallclock time의 진행속도와 맞추기 위해 Sleep함수를 사용한다. 이것은 일반적으로 프로그램 시뮬레이션의 한 주기 실행시간이 매우 빠르기 때문에 시뮬레이션 시간을 실시간 진행속도와 맞추기 위해서 동기화를 위한 실시간 시간의 간격만큼 시간을 지연시켜주기 위해서이다. 또한 그림 7의 Pseudo 코드는 time step을 1로 설정하고 1초 간격으로 동기화를 진행한

```
LT=0
/* This is abstract time. But we assume that this is 1sec for real-time
simulation*/
timeStep = 1

while (SimulationEnd != TRUE) do
        /* Time advance part */
        LT += timeStep
        rtiAmb.timeAdvanceRequest(LT)
        while (TimeGranted != TRUE) do
              //use "tick()" wait callback from RTI.
              rtiAmb.tick()
        end while

        /* Time synch(between logical time and wallclock time) part */
        currWT = getCurrTime()     //time unit is ms
        Sleep(1000 – (currWT-prevWT))
        prevWT = currWT
end while
```

그림 7. Clock Federate의 Pseudo 코드
Fig. 7. Pseudo code of Clock Federate

경우를 보여준다.

그림 8은 LVC GW의 전체 프로세스 동작 순서를 나타낸다.

그러므로 위의 GW Federate와 Clock Federate의 연동을 통해서 LVC GW는 HLA Part의 다른 Federate의 시간 진행을 DDS와 실시간 진행 속도와 동기를 맞출 수 있을 뿐 아니라 DDS로부터 전달된 데이터 메시지를 HLA 파트로 일관성 있게 전달할 수 있도록 해준다.

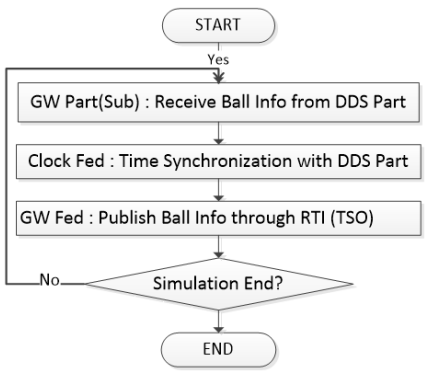

그림 8. LVC GW 프로세스 동작 순서
Fig. 8. Flow chart of LVC GW process

Ⅳ. LVC 연동 모델 및 시간 동기화 검증 환경 설계

그림 9는 HLA/RTI Time management와 Clock Federate를 이용하여 DDS의 실시간에 HLA Federate들의 시뮬레이션 시간이 동기화 되는 과정을 보여주기 위한 LVC 시스템 구성도이다.

L 체계를 시연하기 위해 한 PC에서는 DDS participant를 실행하고, V와 C 체계를 시연하기 위해

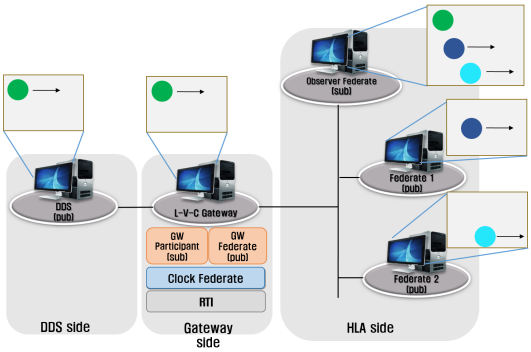

그림 9. LVC GW 시간 동기화 테스트 시스템 모델
Fig. 9. LVC GW time synchronization test system model

3개의 HLA Federate를 각각의 PC에서 동작시켰다. LVC GW PC에서는 DDS Participant (subscriber), GW Federate(publisher) 그리고 Clock Federate를 실행하였다. 개발 프로그램으로는 DDS Participant를 구현을 위해 RTI DDS를 사용하였고, HLA Federate 구현을 위해 MAK RTI를 사용하였다.

시뮬레이션이 시작되면, DDS Side와 HLA Side에서 Publisher로 설정된 DDS Participant와 HLA Federate들은 각각 다른 색깔과 높이(y좌표)에서 볼이 이동하는 것을 시뮬레이션하면서 이와 동시에 자신의 색깔과 위치 정보(x,y좌표)를 메시지에 실어서 Publish하여 Subscriber에게 전송한다. 이때 LVC GW의 경우에는 GW Participant를 통해서 DDS가 publish한 볼 데이터 정보를 수신하여 시뮬레이션 화면에 나타내고 이와 동시에 받은 볼 정보를 DDS-HLA매핑을 통해 변환한 후, GW Federate를 통해서 HLA Side로 Publish한다. 그러면 HLA Side의 Subscriber로 설정된 PC는 HLA Side의 Publisher들을 통해서 받은 2개의 볼 정보와 함께 GW를 통해 받은 볼 정보까지 총 3개의 볼의 정보를 수신하여 화면에서 확인하도록 한다.

또한, LVC 연동 시연에서의 GW를 통한 DDS와 HLA 시뮬레이션의 시간 동기화를 효과적으로 보여주기 위해 HLA Federate로 구현된 볼의 x좌표를 각 시뮬레이션 시간과 일치시켰다.

즉, 시뮬레이션에서 볼은 수평으로 이동하기 때문에 그림 10에서 y_j와 y_{j+1}의 값은 같게 되어 볼의 속도는 아래 공식 (2)과 같다.

$$v_x(i) = \frac{(x_{j+1}(i) - x_j(i))}{(W_{j+1}(i) - W_j(i))} \quad (2)$$

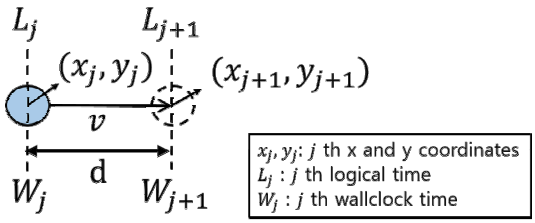

그림 10. 시뮬레이션 Ball의 위치와 진행속도
Fig. 10. Position and speed of Simulation ball

여기서 i는 ball을 구분하기 위한 index로서 사용되고, DDS Participant 혹은 HLA Federate가 될 수 있다. 따라서 위의 두 공식을 통해 x좌표를 시뮬레이션 시간 L과 일치시키면 $v_x(i)$ 값은 $v_L(i)$과 같게 되어 x축 이동 속도를 시뮬레이션 진행속도로 볼 수 있음을 확인할 수 있다.

그러므로 위와 같은 설정을 통해 서로 다른 시스템에서 시뮬레이션 하는 볼의 위치 정보는 시간동기화가 맞지 않으면 다른 위치(x좌표)에서 진행을 하게 될 것이고, 만약 서로 시뮬레이션 시간 진행 속도가 일치하게 된다면, 여러 볼들이 일직선 형태로 진행할 것을 예측할 수 있다.

다음으로 ball을 표현하기 위한 DDS Participant와 HLA Federate들의 기본 설정들은 표 1과 같다.

따라서 HLA Federate들의 x좌표 이동 단위도 8pixel로 고정하고 주기만 다르게 하여 두 개의 Federate들의 시뮬레이션 시간 진행 속도가 DDS 기반 실제 시간진행속도보다 빠른 경우를 가정하였다. GW Federate의 경우, DDS로부터 전달받은 데이터는 바로 맵핑하여 전달해야하기 때문에 DDS와 동일한 속도로 설정하였다.

또한, 위에서 시뮬레이션 시간을 x축 값으로 사용하기로 했기 때문에 모든 Federate들의 time step을 8로 세팅하여 시뮬레이션 시간이 8씩 증가하게 하고, Clock Federate의 경우에는 실행 주기를 80ms로 세팅하여 DDS와 시간 진행 속도를 맞추도록 하였다.

표 1. LVC 시스템 모델의 ball 표현을 위한 기본 설정 값
Table 1. Default setting value for LVC simulation ball

Index(i)	Ball	Mode	Speed (Distance /period)
0	DDS Part	Pub	8pixel/80ms
1	GW Fed	Pub	8pixel/80ms
2	HLA Fed1	Pub	8pixel/50ms
3	HLA Fed2	Pub	8pixel/40ms
4	HLA Fed3	Sub	N/A

V. 실험 및 결과 분석

시뮬레이션 테스트를 위해서 우리는 Federate들에 대하여 두 가지 경우의 Time management 설정 값들을 적용하였다.

표 2는 첫 번째 Time management 설정 값으로, HLA domain의 모든 Federate들의 Time management기능을 disable하였다. 이 경우 RTI는 서로 다른 Federate들의 시간 진행에 관여하지 않게 되고, 각 Federate는 자신의 기본 주기마다의 시간 진행을 할 수 있기 때문에 기본 주기가 짧은 Federate가 더 빨리 시간 진행을 할 수 있게 된다.

그림 11은 첫 번째 Time management를 적용한 후 시뮬레이션을 수행한 것으로, DDS participant와 HLA Federate 3의 실행화면을 캡처한 것과 각 Federate들의 이동거리를 측정하여 비교한 것이다.

DDS Participant는 기본 설정 주기에 따라 ball의 위치 정보를 Publish한다. GW는 자신의 Participant를 통해 이를 Subscribe하고 받은 정보를 그대로 GW Federate를 통해서 HLA domain에 Publish한다. 이때 GW Federate는 Regulating옵션이 Enable되어 있으므로 볼의 위치정보와 자신의 logical time을 이용하여 TSO 메시지를 전송하게 된다. HLA Federate 3는 Subscribe로만 설정되어 있기 때문에 다른 모든 Publish하는 Federate가 어떻게 진행되는지 볼 수 있다.

진행 거리(D)의 관계를 통해서 GW Federate는 DDS Domain과 거의 같은 속도로 진행하고 있음을 알 수 있고, HLA Federate 1과 2는 더 빠른 속도로 진행하고 있음을 알 수 있다. 시간에 대한 이동 거리 값을 계산해보면, 그림 12의 그래프와 같은 결과를 얻을 수 있었다. 또한 이 결과는 표 2에서 설정한 Federate들의 기본 주기에 따라 계산된 이동거리 값과 같음을 알 수 있다.

이 결과를 통해 알 수 있는 것은 GW Federate가 TSO 메시지를 주고 Clock Federate가 시간을 DDS 시간과 맞춰서 진행을 할지라도 HLA 파트의 Federate들이 Time constrained로 설정되어 있지 않으면, Clock Federate의 영향을 받지 않기 때문에 시간 동기화를 이룰 수 없고, TSO 메시지를 받을 수 없기 때문에 인과성을 보장할 수 없다는 것이다.

표 3은 두 번째 Time management 설정 값을 나타낸다. Publish로 설정된 Federate 1과 2는 자신의 ball 위치를 Subscribe로 설정된 Federate 3에게 시간에 맞게 전송해주기 위해 Time-regulating으로 설정되어 있고, 동시에 Clock Federate의 시간 진행에 영향을 받고 또한 GW Federate로부터 전송되는 TSO 메시지를 수신하기 위해 Time-constrained로 설정되었다. 또한, Federate 3의 경우, ball 정보를 업데이트할 필요가 없기 때문에 Time-regulating은 disable하였고, 다른 Federate들의 TSO 메시지로 전달된 ball의 위치 정보를 수신하기 위해 Time-constrained로 설정되었다.

표 2. 첫 번째 Time management 설정
Table 2. First time management setting

Federate	Time management	
	Regulating	Constrained
GW Fed	Enable	Enable
Clock Fed	Enable	Disable
HLA Fed1	Disable	Disable
HLA Fed2	Disable	Disable
HLA Fed3	Disable	Disable

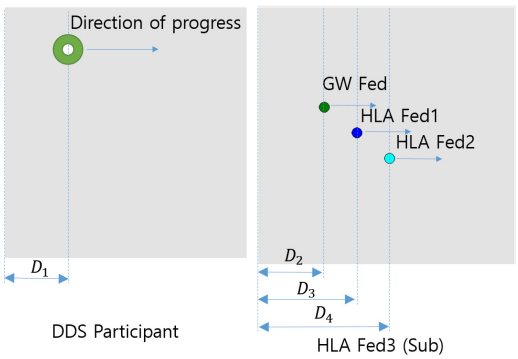

그림 11. 첫 번째 설정에 따른 시뮬레이션 결과
Fig. 11. Simulation result according to first setting

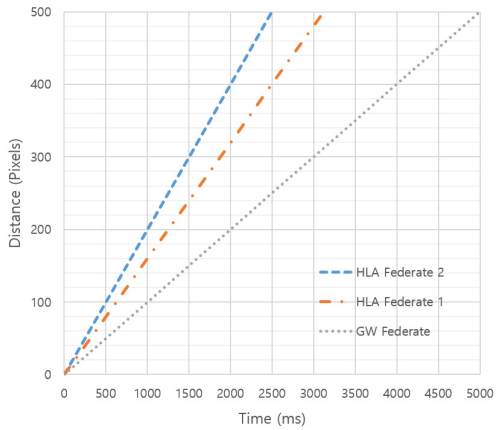

그림 12. 첫 번째 설정에 따른 시뮬레이션 결과
Fig. 12. Simulation result according to first setting

표 3. 두 번째 Time management 설정
Table 3. Second time management setting

Federate	Time management	
	Regulating	Constrained
GW Fed	Enable	Enable
Clock Fed	Enable	Disable
HLA Fed1	Enable	Enable
HLA Fed2	Enable	Enable
HLA Fed3	Disable	Enable

그림 13은 표 3의 설정에 따른 시뮬레이션 결과 화면을 캡처한 것과 각 Federate들의 이동 거리 관계를 나타낸 것이다. DDS Participant와 모든 Federate들의 이동거리가 거의 같음을 알 수 있고, 이는 볼의 진행 속도 즉, 시뮬레이션의 진행 속도가 동일하게 맞춰졌음을 나타낸다. 이것은 각 Federate들이 자신의 기본 주기 설정대로 다른 시간에 RTI에 시간 진행을 요청(TAR)할지라도 RTI는 모든 Time regulating Federate들의 시간 진행을 고려하여 시간 진행을 허락(TAG)하기 때문에 실시간에 맞춰 진행되는 Clock Federate의 시간 주기에 맞게 다른 모든 Federate의 시간이 진행되고, Subscriber 노드에서는 또한, 모든 Publisher Federate 이 자신의 Logical time을 이용하여 전송한 TSO 메시지를 수신하기 때문에 Publisher Federate들이 동일한 시간 진행 속도로 진행한 것을 확인할 수 있다.

이 결과는 DDS 파트에서 시뮬레이션 되는 볼의 위치 정보 업데이트 시간에 맞춰서 HLA 파트의 시뮬레이션이 이루어지는 것과 DDS에서 전송되는 데이터를 인과성 있게 HLA 파트로 전송될 수 있음을 보여준다.

그림 14를 통해서도 모든 Federate들의 ball의 속도 즉, 이동 시간에 따른 이동 거리가 화면의 마지막에 도달할 때까지 동일함을 볼 수 있고 이를 통해 모든

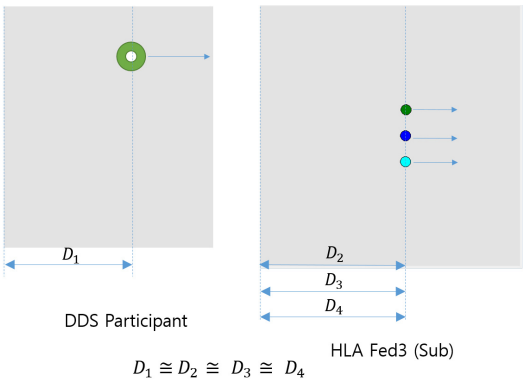

그림 13. 두 번째 설정에 따른 시뮬레이션 결과
Fig. 13. Simulation result according to second setting

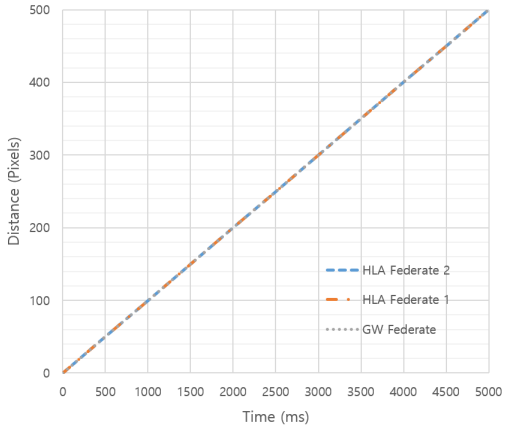

그림 14. 두 번째 설정에 따른 시뮬레이션 결과
Fig. 14. Simulation result according to second setting

Federate들의 시뮬레이션 시간의 진행 속도가 DDS의 실시간 진행속도에 맞춰졌음을 알 수 있다. 또한, 이것을 통해서 DDS의 시간 진행 속도에 Federate들이 동기화되었을 뿐 아니라 HLA Federate들(Fed1, Fed2) 간에도 서로 동기화가 이루어졌음을 알 수 있다.

Ⅵ. 결 론

LVC 통신 미들웨어에서 시간 동기화는 매우 중요한 문제이다. L 시스템의 시간은 Real time으로 동작되고 Real time의 시간 진행은 조절할 수 없다. 따라서 시뮬레이션으로 동작되는 V, C 체계의 시뮬레이션 시간을 Real time의 진행 속도에 맞추는 것을 통해 LVC 연동 시스템의 시간 진행을 맞출 수 있다.

본 논문에서는 L 체계의 미들웨어로 사용되는 DDS의 Real time에 V, C 체계의 미들웨어로 사용되는 HLA의 시뮬레이션 시간을 동기화하기 위해 HLA의 Time management와 실시간에 동기를 맞춰 시간을 진행하는 Clock Federate를 이용한 기법을 제안하였다.

또한, DDS와 HLA를 사용하여 볼 시뮬레이션 테스트를 통해서 LVC GW에서 동작하는 LVC 미들웨어에 Clock Federate와 HLA의 Time management기능을 사용함으로써 HLA의 Federate들이 DDS의 실시간에 동기화되어 진행할 수 있을 뿐 아니라 DDS 파트에서 전송되는 메시지(이벤트)들을 인과성 있게 HLA 파트로 전송될 수 있음을 확인하였다.

본 논문에서 수행한 시뮬레이션 결과를 통해 다른 체계의 시뮬레이션이 혼재된 LVC 통합 시뮬레이션을 위한 LVC연동 미들웨어를 개발함에 있어 HLA Time

management와 Clock Federate 기능을 이용한 이종 체계 시간 동기화 기법을 활용한다면 이벤트의 인과성을 유지하면서 정밀한 시간 동기화가 요구되는 통합 시뮬레이션을 개발하는데 있어 큰 기여를 할 수 있을 것으로 기대된다.

References

[1] J. H. Hong, K. M. Seo, and T. G. Kim, "Reverse simulation software architecture for required performance analysis of defense system," *J. KICS*, vol. 40, no. 4, pp. 750-759, Apr. 2015.

[2] S. Kang, I. Chun, J. Park, and W. Kim, "Model-based autonomic computing framework for cyber-physical systems," *J. IEMEK*, vol. 7, no. 5, pp. 267-275, Oct. 2012.

[3] H. J. Park, "The vision of LVC Integrated Drill for Korean Army," *J. Defense Sci. & Technol. Inf.,* pp. 6-7, 2015.

[4] H. S. Kim, C. S. Jeong, M. H. Lee, B. Y. Seong, S. W. Choi, M. S. Choi, and Y. K. Kim, "A real time synchronous V - C system with the extracted data from buffering function," *ICN 2013*, pp. 175-183, Seville, Spain, Jan. 2013.

[5] P. M. Gustavsson, U. Björkman, and J. Wemmergård, "LVC aspects and integration of live simulation," *Fall Interoperability Workshop*, Orlando, U.S.A, Sept. 2009.

[6] W. C. Choi, K. H Yu, B. J. Park, S. J. Kang and J. H. Lee, "Study on L-V-C(Live-Virtual-Constructive) interoperation for the national defense M&S(Modeling & Simulation)," *Int. Conf. Inf. Sci. and Security*, pp. 128-133, Seoul, Korea, Jan. 2008.

[7] S. Kang, M. Kim, J. Park, I. Chun, and W. Kim, "LVC-Interoperation development framework for acquiring high reliable cyber-physical weapon systems," *J. KICS*, vol. 38C, no. 12, pp. 1228-1236, Dec. 2013.

[8] OMG Technical Document, *Data Distribution Service Specification for Real-Time Systems*, Version 1.2, OMG, Jan. 2007.

[9] Y. J. Song, "Performance analysis of DDS for distribution network management system suitable for satellite communication," *J. KICS*, vol. 38C, No. 12, pp. 1179-1185, Dec. 2013.

[10] IEEE, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules*, IEEE Std. 1516-2000, Aug. 2010.

[11] IEEE, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification,* IEEE Std. 1516.1-2000, Aug. 2010.

[12] IEEE, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT),* IEEE Std 1516.2-2000, Aug, 2010.

[13] Defense Modeling and Simulation Office (DMSO), *High Level Architecture Run-Time Infrastructure RTI 1.3NG Programmer's Guide* Version 6.4.3(1999), from https://hla-rti.wikispaces.com/.

[14] R. M. Fujimoto, *HLA Time Management: Design Documents Version 1.0(1996)*, from http://www.cc.gatech.edu/computing/pads/papers.html#1996.

[15] J. B. Chaudron, D. Saussié, P. Siron, and M. Adelantado, "Real-time distributed simulations in an HLA framework: Application to aircraft simulation," *J. Simulation : Soc. for Comput. Simulation Int.*, vol. 90, no. 6, pp. 627-643, Jun. 2014.

[16] F. Kuhl, R. Weatherly and J. Dahmann, *Creating computer simulation systems : an introduction to the high level architecture*, Prentice Hall PTR, Oct. 1999.

홍 석 준 (Seok-Joon Hong)

2006년 2월 : 한양대학교 미디어통신공학과 학사
2008년 8월 : 한양대학교 전자컴퓨터통신공학과 석사
2008년~2010년 : 인포뱅크 텔레매틱스 부서 연구원
2011년 3월~현재 : 한양대학교 전자컴퓨터통신공학과 박사과정
<관심분야> 스마트그리드, 시뮬레이션, IoT

조 인 휘 (In-Whee Jeo)

1983년 2월 : 한양대학교 전자공학과 학사
1995년 2월 : University of Arizona 정보통신 석사
1998년 2월 : Georgia Tech 정보통신 박사
1985년~1992년 : ㈜ 데이콤 종합연구소 선임연구원
1998년~2000년 : Oak Ridge 국립연구소 연구원
2000년~2002년 : Bellcore Lab(Telcordia) Scientist
2002년~현재 : 한양대학교 컴퓨터공학부 교수
<관심분야> 이동통신, IoT, 멀티미디어 네트워크

이 우 엽 (Woo-Yeob Lee)

2007년 2월 한양대학교 미디어통신공학과 학사
2009년 2월 : 한양대학교 전자컴퓨터통신공학과 석사
2009년 9월~현재 : 한양대학교 전자컴퓨터통신공학과 박사과정
<관심분야> 이동통신, IoT, 미들웨어

김 원 태 (Won-Tae Kim)

1994년 2월 : 한양대학교 전자공학과 학사
1996년 2월 : 한양대학교 전자공학과 석사
2000년 8월 : 한양대학교 전자공학과 박사
2001년~2005년 : ㈜로스틱테크놀로지 기술이사
2005년 3월~2015년 8월 : 한국전자통신연구원 CPS 연구실 실장/책임연구원
2015년 9월~현재 : 한국기술교육대학교 컴퓨터공학부 조교수
<관심분야> CPS, IoT, Digital twin, Modeling& Simulation