Vol.5, No.11 pp.393~402 pISSN: 2287-5891

SIA-LVC: Scalable Interworking Architecture for Military L-V-C Training Systems Based on Data Centric Middleware

Won-Tae Kim[†] · Seung-Min Park^{††}

ABSTRACT

A Military L-V-C system consists of distributed complex systems integrating Live systems working on physical wall-clock time, Virtual systems ruled by virtually pseudo realtime events on a computer, and Constructive systems only depending on the causal relationship between the continuous events. Recently many needs for L-V-C training systems are increasing in order to achieve the maximum training effects with low costs. While theoretical/logical researches or only partially interworking technologies have been proposed, there are few perfect interworking architectures for totally interoperating L-V-C systems in world-wide. In this paper, we design and develop a novel interworking architecture based on data centric middleware for the consistent global time with the same states on the entire L-V-C data and events by means of integrating the heterogeneous distributed middleware standards of each L-V-C system. In addition, simulated L-V-C systems based on real systems will be used for the efficiency and performance of the developed interworking architecture.

Keywords: L-V-C, Scalability, Data Centric Middleware, Global Data Space, Gateway

SIA-LVC: 데이터 중심 미들웨어 기반 확장성 있는 국방 L-V-C 훈련체계 연동 아키텍쳐

김 원 태^{*}·박 승 민^{**}

요 약

국방 L-V-C 시스템은 물리적 시간축에 따라 이벤트가 진행되는 Live 시스템, 컴퓨터 상에서 실제시간에 근접한 시간 사건에 의해 지배되는 Virtual 시스템 및 진행 시간에 관계없이 사건간 인과관계에만 의존적인 Constructive 시스템 등이 혼재된 분산형 복잡 시스템이다. 최근 이들 훈련 시스템들을 연동하여 최적의 훈련효과를 얻고자하는 LVC 연동 훈련체계에 대한 요구가 전세계적으로 증가하고 있다. 그러나, 기존에는 이론적이고 논리적인 접근 방식 혹은 부분적인 연동만이 제한적으로 제안되어 온 반면, 전 시스템적으로 LVC 훈련체계들을 연동시킬 수 있는 실제적인 기술은 국내외적으로 드문 상황이다. 이에 본 논문에서는 각 훈련시스템의 고유한 특성을 지원하는 분산시스템 연동 프로토콜들을 상위 개념에서 통합하고, 데이터와 이벤트에 대해 동일한 글로벌 시간과 상태를 유지하기 위한 데이터 중심 미들웨어 기반의 새로운 연동 아키텍처를 설계하고 구현한다. 또한, 구현된 연동 아키텍처를 기반으로 실제 L-V-C 시스템들을 모사한 시연 시스템들을 활용하여 그 성능을 검증하고 유효성을 증명한다.

키워드: L-V-C, 확장성, 데이터 중심 미들웨어, 글로벌 데이터 공간, 게이트웨이

1. 서 론

1.1 연구의 배경

국제적으로 국방기술의 발전과 국방비 절감 요구에 따라 전장 내 모든 군, 무기체계, 병사 등을 통합하여 운용하는 네트워크중심전으로 패러다임이 전환되었다. 이는 네트워크 를 기반으로 전장에서 발생하는 모든 정보를 실시간적이며

[※] 본 연구는 산업통산자원부와 방위사업청 주관의 민군기술협력진홍센터의 지원으로 수행되었음.

[※] 이 논문은 2015년도 한국기술교육대학교 신임교수 연구과제 연구비 지원에 의하여 연구되었음.

^{*} 정 회 원:한국기술교육대학교 컴퓨터공학부 조교수

^{**} 정 회 원: 한국전자통신연구원 임베디드SW연구부 책임연구원 Manuscript Received: August 24, 2016 Accepted: September 19, 2016

^{*} Corresponding Author: Won-Tae Kim(wtkim@koreatech.ac.kr)

효과적으로 공유하고, 더 나아가 경제적으로 전쟁을 수행할수 있도록 한다. 최근에는 대규모 전투훈련체계로서 실제적인 전투체험과 각종 전략모의 등을 경험할수 있으며, 이종훈련체계들간 정보를 실시간 공유하고 전체 시스템 수준에서의 상태일관성을 동일하게 유지하는 방안에도 활용되고 있다. 이러한 훈련체계를 L-V-C(Live-Virtual-Constructive)훈련체계라 한다[1, 10]. 특히, 완벽하게 구성된 L-V-C 훈련체계를 구축하기 위해서는 개별적으로 운용되는 훈련체계들에서 발생되는 이벤트와 데이터 및 개별 통신 프로토콜들의상호 연동 기술이 핵심이다[11]. 따라서, 각 훈련시스템별로개별적 운용되던 표준적인 분산 미들웨어를 수용하면서 확장성있는 연동을 위한 통신 미들웨어 프레임워크를 구축하고 이를 기반으로 L-V-C 훈련체계들을 일관성 있게 융합하는 체계적 방안이 필요하다.

1.2 L-V-C 훈련체계 특징

실제(Live) 훈련체계인 강원도 인제에 위치한 KCTC(육 군과학화전투훈련단)에서는 레이저건이 장착된 마일즈장비 를 탑재한 개인화기 및 무기체계들을 활용하여 수천명의 훈련병들을 동시에 훈련시킬 수 있는 첨단 Live 훈련장이 있다. 일반적으로 Live 훈련체계들에서는 초당 수만건의 이 벤트가 동시에 발생하게 되고, 부가 정보데이터를 포함하면 지속적인 대규모 데이터가 흘러 다니는 실시간 시스템이라 할 수 있다. 가상(Virtual) 훈련체계는 주로 무기체계에 대 한 사용법 훈련을 위한 체계로 전투기나 전차와 같은 복잡 한 중장비를 운용하기 위한 시뮬레이터기반 훈련체계이다. 통상 가상 훈련체계들은 시뮬레이터 연동 통신 미들웨어를 통해 시뮬레이터에서 발생하는 이벤트를 분산 시뮬레이션 용의 통신미들웨어를 기반으로 공유하는 구조를 취한다[7]. 구성 (Constructive) 훈련체계는 가장 낮은 해상도 혹은 정 밀도를 갖는 워게임 수준의 훈련체계이다. 이는 주로 지휘 관 훈련에 활용되며, 대규모 작전실행 훈련을 비실시간적으 로 진행하게 되고 확률/통계이론에 근거하여 병력간 교전 시 교전결과가 결정된다. 현재까지의 구성 훈련체계는 지휘 관의 교전 명령에 대응하여 워게임엔진(예, 전투21)에 운용 자가 입력을 함으로써 워게임이 진행되는 형태이다. 따라 서, 비실시간적이며, 실제 무기체계의 특성과 병사의 심리 적 상태 등 다양한 현실적 상태가 훈련 시뮬레이션에 반영 되지 못한다. 서로 다른 특성을 갖는 상호보완적 시뮬레이 션 훈련체계인 L-V-C 훈련체계들을 실시간 연동하여 종합 적인 군사훈련이 가능한 체계를 구축하는 것이 최근 국방 훈련체계의 중요한 이슈로 등장하고 있는 바, 본 논문에서 는 이종 훈련체계들을 상호연동함으로써 전장에서의 이벤 트를 실시간 공유하여 효과적인 훈련이 가능하도록 하는 L-V-C 연동 지원 통신 미들웨어에 대한 관련 연구, 시스 템 요구사항, 연동 아키텍쳐 설계 및 구현, 그리고 실험 및 분석 등을 각 절에서 설명하도록 한다.

2. 관련 연구

L-V-C 혹은 부분적인 연동을 지원할 수 있는 대표적인 연동 미들웨어들과 본 논문에서 제안하는 SIA-LVC간 정성 적 비교를 이후 상세한 설명에 앞서 Table 1에 제시한다.

Table 1. The Functional Comparison Between L-V-C Interworking Middleware Candidates

| 항목 | DDS[8] | HLA/RTI [2, 3] | RT-COR BA[5] | TENA[6] | SIA-LVC |
|--------------|--------------|-------------------|-----------------|------------------------------------|--------------------------|
| 데이터 전송 | Pub/Sub | RTI 제어 | Broker | Broker + Pub/Sub | Pub/Sub |
| 통신 구조 | 완전 분산 | 중앙 집중 | 중앙 집중 | 중앙 집중 | 완전 분산 |
| 세션 지원 | Domain 기반 | Federation 기반 | NA | Range 기반 | Global Federation |
| 통신 주체 | Participant | Federate | Object | Stateful Distribute d Object | Participant /Federate |
| 실시간성 | Soft RT | 비실시간 | Hard RT | Hard RT | Soft RT |
| 시간동기화 | NA | 강력 지원 | NA | NA | 강력 지원 |
| QoS | 22개 QoS | 안전성, 순서보장 | 실시간성 만 지원 | 실시간성 만 지원 | 22개 QoS |
| 동적참여 | 참여 가능 | 불가 | 참여 가능 | 참여 가능 | 참여 가능 |
| 확장성 | 중간 | 낮음 | 낮음 | 낮음 | 높음 |
| 동시지원 노드수 | 중규모 (<수백) | 소규모 (<수십) | 중규모 (<수백) | 중규모 (<수백) | 대규모 (>수천개) |
| 이종미들웨어 연동 | NA | NA | NA | GW | GW |
| 타켓 시스템 | 실시스템 (운용) | 시뮬레이터 | 실시스템 (컴퓨팅) | 실시스템 (테스트) | L-V-C 전체 |

3. SIA-LVC 아키텍쳐 설계

3.1 분산 L-V-C 통신미들웨어 프레임워크 요구사항

앞서 정의한 바와 같이 L-V-C 훈련체계를 구성하는 각 L/V/C 훈련체계 요소들은 각각의 독립적인 필요성과 이를 지원하는 시스템 요구사항을 기반으로 개발되었다. 본 연구에서의 L-V-C 훈련체계 통합이란 각 요소들의 기본적인 동작성을 보장하면서 동시에 전체적인 L-V-C 통합 훈련체계기능을 제공하는 것을 목표로 한다. Fig. 1에서는 L-V-C 통신 미들웨어가 지원해야하는 요구사항을 기존의 국내외 부분적으로 개발되어 운용된 L-V-C 시스템의 문제점들을 분석하고, 국방 분야 전문가들의 도움을 기반으로 기술적 요구사항을 도출하였다.

주요 요구사항으로는 대규모 L-V-C 객체들이 동시에 연동될 수 있는 환경 지원, 통신 객체들간의 안정적 데이터 공유, 각 L/V/C 체계들에서 기 적용하고 있는 통신 미들웨어 및 데이터 수준에서의 연동 혹은 변환기능의 제공, 인터넷과 같은 WAN 환경으로의 확장성 지원, 전 L-V-C 훈련체계 내에서의 세션 혹은 페더레이션 관리 및 시간축이 다

른 가상훈련체계와 실제 훈련체계간의 시간 동기화 제공 등 이다. 다만, 시뮬레이션 다중해상도 지원, 시뮬레이션 오브젝 트 모델 지원, 합성전장환경 공유 등은 L-V-C 훈련체계에 서 매우 중요한 기능이나 시뮬레이션 응용계층의 기능 요구 사항이므로 본 연구 대상에서 제외한다.

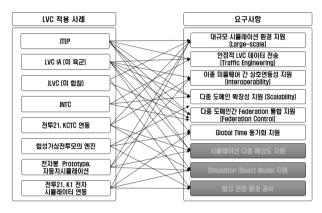


Fig. 1. Technical Requirements of L-V-C Training Systems

3.2 SIA-LVC 아키텍쳐

Fig. 1에서 도출된 L-V-C 훈련체계의 기술적 요구사항을 만족하는 L-V-C 아키텍쳐를 Fig. 2에 도시한다. 특히, 광역 대규모성, 안정적 데이터 전송 지원, 이종 미들웨어간 연동 성, 시스템 확장성, 글로벌 패더레이션 관리 및 시간 동기화 기법 등을 제공할 수 있는 아키텍쳐로서 설계되었다. 물리 적 인터넷 환경에서 지역적으로 분산되어 있는 각 훈련시스 템의 특화 미들웨어 도메인들을 다양한 구조의 데이터 배포 에 최적화된 DDS를 기반으로 하나의 가상 도메인 즉 글로 벌 데이터 공간(Global Data Space)으로 통합하는 구조를 취한다. 그러나, 표준 DDS는 로칼 네트워크만을 지원하고 있어, 광역 대규모성 및 확장성을 제공하기 위한 릴레이 노 드가 요구된다. 릴레이 노드들은 이종 미들웨어에 의해 제 어되는 각 L/V/C 훈련체계 내 객체들을 글로벌 데이터 스 페이스에서 투명하게 인식될 수 있도록 지원한다. 또한, 표 준 DDS는 개별 지역 도메인들에서 발생하는 데이터들의 안 정적 전송을 제공하므로, 이를 통해 대규모 객체들로부터 발 생하는 데이터의 신뢰성을 보장할 수 있다. SIA-LVC 릴레 이 노드는 원격 객체를 로칼 내의 객체로 가상화하기 위한 가상 PDP 관리 블록(VPM: Virtual Participant Management Block), 라우터 연결 관리 블록(RCM: Router Connection Management Block), 실제 통신 객체인 엔터티의 가상화 블 록(VEM: Virtual Entity Management Block) 및 글로벌 토 픽 관리 블록(Global Topic Management Block)으로 구성된 다. 인바운드 통신 채널은 DDS를 통해 SIA-LVC 게이트웨 이와 연동되고, 인터넷으로의 아웃바운드는 릴레이 노드가 관리하는 TCP 연결을 통해 다른 릴레이 노드들과 연결된다.

SIA-LVC 게이트웨이는 다양한 통신 미들웨어를 지원한 다. 이는 각 L/V/C 훈련체계들을 지원하는 연동 미들웨어들 은 프로토콜 및 데이터 구조체가 다르며, 훈련체계의 응용

계층에서 정의하는 객체 구조체도 상이하기 때문이다. 현재 세계적인 L/V/C 각 훈련체계에서 활용중인 미들웨어로는 DIS(Distributed Interactive Simulation)와 IEEE HLA/RTI 가 가상훈련체계(V)와 워게임 구성체계(C)에 가장 많이 사 용되며, 실체계(L)에는 DDS가 지배적이다[2, 4, 8]. 다만, DIS는 미들웨어의 기능보다는 PDU(Packet Data Unit)으로 서 연동 시스템간의 데이터 헤더 포맷만을 정의한다. 본 논 문의 SIA-LVC 게이트웨이는 구조적으로 다양한 미들웨어 를 지원할 수 있도록 온톨로지기반 변환구조를 취하도록 설 계한다.

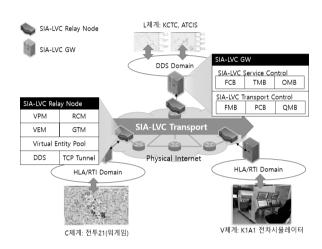


Fig. 2. SIA-LVC Architecture

SIA-LVC 전송제어계층에서는 전송계층의 프로토콜을 상 호변환시키는 프로토콜 변환 블록(PCB: Protocol Conversion Block), 상이한 QoS 제공성을 보정하는 QoS 관리 블록(QMB: QoS Management Block) 및 사용자의 FOM(Federation Object Model) 정보를 통해 게이트웨이 내에서 외부와 통신할 개체를 생성하는 FOM 관리 블록(FMB: FOM Management Block) 등으로 구성된다. SIA-LVC 서비스 제어 계층에서는 L-V-C 체계의 글로버 패더레이션의 세션을 관리하는 패더 레이션 제어블록(FCB: Federation Control Block), 시간축이 다른 각각의 L/V/C 체계들간의 시간 동기화를 담당하는 시 간 관리 블록(TMB: Time Management Block) 및 이종 프 로토콜 및 데이터 포맷을 매칭시키는 온톨로지 관리 블록 (OMB: Ontology Management Block)이 유기적으로 연동된 다. 각각의 기능블록에 대해서는 4장에서 구현체 설명과 함 께 상세히 제시한다.

4. SIA-LVC 아키텍쳐 구현

4.1 SIA-LVC 전송 프로토콜

DDS 기술은 분산 환경에서 발간 (Publish)/구독(Subscribe) 모델에 따른 데이터배포를 명세하고 있다. DDS는 응용이 발간이나 구독을 원하는 데이터를 토픽 형태로 정의하고 이 와 관련된 서비스품질(Quality of Service: QoS) 지정하기만

하면 실제 통신을 위한 상대방 통신개체의 탐색, 통신 연결의 설정, 지정된 QoS의 지원 등이 미들웨어 차원에서 응용에 투명하게 제공된다. 또한, 다양한 응용이 생성하는 토픽별 속성에 따른 데이터 품질을 정의하고 이를 반영한 독자적인 발간-구독 통신 방식이 제안되고 있다[8].

DDS를 통해 분산 응용은 그 목표 기능에 따라 다양한 특성을 가질 뿐만아니라 이질적인 환경에서 운용될 가능성이 상존한다. 이를 위해서는 각 응용 및 그 운용환경 하에서 요구하는 데이터 교환 특성을 만족시킬 수 있어야만 한다. DDS는 총 22개의 서비스품질(QoS: Quality of Service) 정책을 지원하도록 하고 있으며, DDS에서 정의되는 발간자, 구독자인 통신 개체 및 교환 데이터인 토픽별로 각기 다른서비스품질을 설정하고 운용할 수 있다.

4.2 SIA-LVC 게이트웨이

1) SIA-LVC Transport Control

a) FOM 관리 블록(FMB)

사용자 FOM 정보를 파성하여 HLA, DIS, DDS의 통신 개체를 생성하고, L-V-C 게이트웨이가 실행될 수 있는 동 작과 관련된 소스코드를 생성하는 기능을 제공한다.

- FOM 파싱 모듈: 표준 RPR FOM을 입력으로 HLA, DIS, DDS에 필요한 자료구조를 만들기 위해서 사용되는 파서[9]
- FOM 관리 모듈: 파싱된 표준 RPR FOM의 정보를 저장하기 위한 자료 구조로, RPR FOM의 객체 (Object) 및 인터랙션(Interaction) 정보와 관련된 속성 정보를 저장 하고 관리하는 기능 제공
- Fed/Topic 생성 모듈: FOM 관리 모듈에 저장된 정보를 바탕으로 DDS 토픽을 생성하고 HLA의 Fed 파일을 생성하는 모듈로, HLA, DDS 구동을 위한 헤더 파일을 생성하는 기능 제공
- 소스 코드 생성 모듈: Fed/Topic 생성 모듈에 의해서 생성된 Topic 정보 및 Fed 파일 정보를 바탕으로 HLA, DDS가 구동되기 위한 실행 소스 코드를 생성 하는 기능을 제공, 여기서는 main문을 바탕으로 실제 HLA, DDS 가 동작 및 제어되기 위한 대부분의 코드가 생성됨

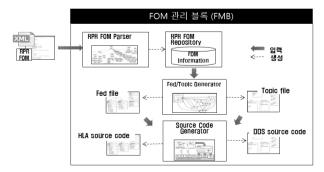


Fig. 3. FOM Management Block Diagram

b) 프로토콜 변환 블록 (PCB)

L-V-C Gateway에서 연동하려고 하는 이종 미들웨어 간데이터 변환 및 전달 기능을 수행하는 기능을 제공하는 블록으로, 현재 본 논문에서는 대표적인 L/V/C 미들웨어인 HLA/RTI와 DDS에 대한 데이터 변환 및 전달에 대한 기능을 명시한다. 특히, 데이터 변환은 기존 미들웨어의 수정없이 데이터 변환이 가능하도록 수정 변경이 최소화 되도록해야 한다.

- 개체 관리 모듈: 데이터 전달 및 변환을 위해서 L-V-C Gateway에서 생성된 HLA의 객체와 DDS 엔터티 등을 관리하는 모듈
- 개체 매핑 모듈: 데이터 전달 및 변환을 위해서 L-V-C Gateway에서 생성된 HLA의 객체, DDS 엔터티를 RPR FOM의 데이터 타입에 따라 상호 연관 관계 정보를 제공해 주는 기능, RPR FOM 정보를 바탕으로 HLA, DDS는 서로 연관 관계가 있는 개체가 생김
- 데이터/API 연동 모듈: L-V-C Gateway에서 생성된 HLA의 객체와 DDS 엔터티간 실제 데이터 전달을 위해서 필요한 함수 및 인터페이스에 대한 기능 제공
- 이벤트 처리 모듈: L-V-C Gateway에서 생성된 HLA 의 객체와 DDS 엔터티에 데이터 도착과 같은 이벤트를 처리하기 위한 Callback 및 이벤트 핸들러 기능 제공

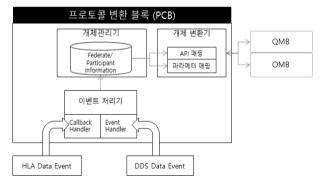


Fig. 4. Protocol Conversion Block Diagram

c) QoS 관리 블록 (QMB)

QMB 블록은 게이트웨이의 HLA FOM DB로부터 QoS Policy의 추출 기능을 제공하는 QEM 모듈과 GD_Entity와 DDS_Entity 사이에 SEDP에 의한 통신 링크 생성 기능을 제공하는 QMM 모듈과 데이터를 송수신할 때, 식별된 QoS Policy의 처리 기능을 제공하는 QPM 모듈로 구성된다.

특히, QoS 처리를 담당하는 QPM의 주요 기능 모듈들은 다음과 같다.

• Reliability 처리 모듈: HLA와 DDS는 데이터 전달과 변환시 데이터의 손실 없이 데이터의 전달과 변환을 수행하는 기능으로 본 모듈은 L-V-C Gateway에서 데 이터의 손실 없이 데이터 전달 및 변환을 수행하는 기 능 제공

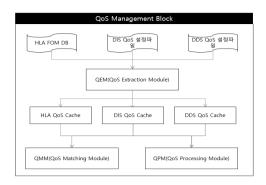


Fig. 5. QoS Management Block Diagram

- Ownership 처리 모듈: HLA와 DDS는 생성된 객체와 엔터티에 대해서 데이터 발생 권한을 가질 수 있는 기 능으로 본 모듈은 L-V-C Gateway에서 생성된 객체와 엔터티에 대해서 데이터 발생에 대한 우선권을 부여하 는 기능 제공
- Time-Ordering 처리 모듈: HLA와 DDS는 Time-Ordering과 관련하여 TSO(Time stepped Ordering)와 RO(Received Ordering)을 제공할 수 있는 기능으로 본 모듈은 L-V-C Gateway에서 생성된 객체(Object)와 엔터티(Entity)에 의해서 생성된 데이터에 대한 TSO와 RO 정보를 제공해 주는 기능 제공
- 데이터 필터링 모듈: HLA와 DDS는 생성된 데이터에 대한 필터링 기능을 제공하는 기능으로 L-V-C Gateway 에서 생성 객체와 엔터티에 대해서 데이터 발생 권한을 가질 수 있는 기능으로 본 모듈은 L-V-C Gateway에 서 생성된 객체와 엔터티에 대해서 데이터 발생에 대 한 우선권을 부여하는 기능 제공

QoS 관리 블록은 HLA/RTI의 4개 QoS와 DDS의 22종 QoS간의 매핑을 Fig. 6과 같이 수행한다.

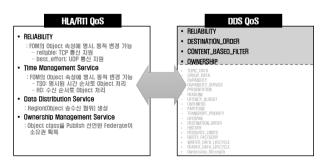


Fig. 6. QoS Conversion Rule Between HLA/RTI and DDS

2) SIA-LVC Service Control

a) 패더레이션 제어 블록(FCB)

FCB는 DDS가 HLA RTI Federation에 참여, 탈퇴, 저장, 동기화를 할 수 있는 기능을 제공하며, 주요 기능 모듈들은 다음과 같다.

- 객체 탐색 모듈(ODM: Object Discovery Module): DDS와 HLA Federate가 서로 통신 객체를 탐색하는 기능 제공
- 객체 변환 모듈(OCM: Object Conversion Module): HLA 통신객체와 DDS 통신객체를 연결하기 위해 오브 젝트 단위로 매핑 및 변환 기능 제공
- 동기화 관리 모듈(SMM: Synchronization Management Module): DDS와 HLA의 실행 시점 동기화 관리 기능
- 회복 관리 모듈(RM: Recovery Management Module): 저장된 시점의 상태와 메시지를 되돌리기 위해 사용하 는 기능 제공

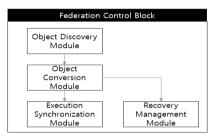


Fig. 7. Federation Control Block Diagram

b) 시간 관리 블록(TMB)

TMB는 DDS와 HLA RTI Federation가 서로 동기화된 시간에 맞추어서 메시지를 시간 순서에 따라 주고받는 기능 제공한다.

- 정책결정모듈(RCPM: Regulation & Constrained Policy Module): 시간에 따라서 메시지를 전달하는 Regulation 모드와 메시지를 수신하는 Constrained 모드를 결정하 는 기능 제공
- 시간 전진 모듈(TAM: Time Advance Module): HLA 와 DDS가 시간을 전진시켜 해당되는 시간에 메시지를 주고받을 수 있는 기능 제공
- 시간 상태 관리 모듈(SPM: Synchronization Point Module) : 시간을 전진시켰을 때 상태를 관리하는 기능 제공
- 시각 변환 모듈(CCM: Clock Conversion Module): Wall clock으로 HLA 시간을 변환하는 기능 제공

특히, TMB에는 클록 페더레이트 (Clock Federate) 개념 을 적용하여 HLA time management를 수행한다. 클록 페더 레이트는 실세계 시간 (Wallclock time)과 가상시스템인 페 더레이트 실행 진행 속도를 맞추기 위해 사용된다. DDS에 의해 연동되는 Live 시스템들은 wallclock time으로 동작한 다. 클록 페더레이트는 논리 시간 (Logical time)을 실세계 시간과 동기화하여 진행한다. 즉, 클록 페더레이트의 논리 시간은 실세계 시간과 정확히 일치된다. 따라서, 나머지 일 반 페더레이트들은 클록 페더레이트와 동기를 맞춤으로써 전체 페더레이션 내의 모든 페더레이트들은 실세계 시간과 동기화되어 동작하게 된다.

c) 온톨로지 관리 블록(OMB)

OMB는 L-V-C 연동 미들웨어에 다양한 이종 미들웨어 (DDS, HLA 및 DIS)의 이식성을 지원하기 위해 온톨로지모델링을 통하여 각 이종 미들웨어의 API 변환과 데이터 변환 기능을 제공하며, 주요 기능 모듈로는 다음과 같다.

- API 매핑 모듈(AMM: API Mapping Module): 입력된 API와 동일한 기능을 수행하는 이종 미들웨어의 API를 맵핑하여 검색하는 기능 제공
- DCM 매핑 모듈(DMM: Data Mapping Module): 입력된 미들웨어이 데이터 타입과 동일한 데이터 타입을 맵핑하 여 변환하는 기능 제공

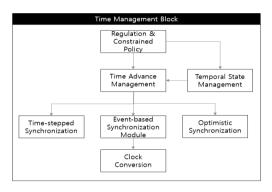


Fig. 8, Time Management Block Diagram

AMM 모듈은 온톨로지 모델을 기반으로 이종 미들웨어 DDS와 HLA API에 대해서 SameAs라는 관계를 통해 매핑 가능한 API를 도출하는 역할을 한다. AMM 모듈은 온톨로지 모델링 저장소에 저장되어 있는 API 맵핑 테이블로부터 SPARQL Query를 이용하여 맵핑되는 API를 추출하는 기능을 수행한다.

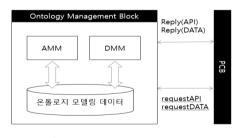


Fig. 9. Ontology Management Block Diagram

DMM 모듈은 DDS의 데이터 타입을 정의하는 IDL을 기반으로 이를 XML로 변환하고 이를 이용하여 이종 미들웨어의 데이터 타입과 맵핑되는 데이터를 DBMS에 저장, 관리하는 기능을 수행한다.

4.3 SIA-LVC 릴레이 노드

릴레이노드는 광역 인터넷 환경에서 글로벌 테이터 공간을 구축할 수 있는 네트워크 연계 기능을 제공한다. 이를 Fig. 10에 도시한다.

1) 가상 PDP 관리자(VPM: Virtual PDP Manager)

가상 PDP관리를 위한 순서는 Fig. 11과 같다. 새로운 릴 레이 노드에 접속하거나 접속 요청을 받게 되면, 먼저 접속 요청한 릴레이 노드의 PDP GUID를 전송한다. 그렇지 않을 경우, 동시에 접속이 이루어 지면, 어떤 라우터인지 구별을할 수가 없고, 우선순위 결정도 어렵다. 또한, PDP GUID를 전송함으로써, 이후 설명할 VEM에서의 Virtual Writer 및 Virtual Reader를 생성 시 기준 GUID를 선택 할 수 있다. 접속이 이루어지고, 새로운 PDP GUID를 할당 받으면, 상대 릴레이 노드에게 현재 BuiltinPublisher 및 BuiltinSubscriber의 History cache에 저장되어 있는 cache 정보를 모두 전송하여 데이터를 백업시킨다.

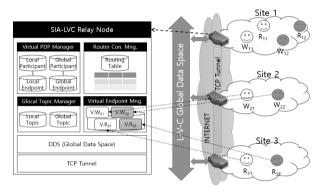


Fig. 10. SIA-LVC Relay Node Diagram

2) 릴레이노드 연결 관리자(RCM: Relay node Connection Manager)

RCM은 동적인 네트워크 구성을 위해 릴레이 노드들 간의 TCP 연결을 구성하기 위한 프로토콜을 정의한다. 이는 WAN에서는 DDS 네트워크가 구축되어 있지 않은 관계로일종의 오버레이 네트워크를 구축하는 방식이다. IP 터널링에 비해 TCP는 안정적 데이터 전송 효과를 얻을 수 있으며, VPM/VEM/GTM과 같은 통신제어 서비스를 제공하므로 RCM을 통해 DDS와 TCP 간을 연계하는 것은 통신 구조상장점이 될 수 있다. TCP 연결설정 후에는 DDS 메시지들을설정된 TCP 채널을 통해 상대 릴레이 노드들로 전송한다. 또한, 헬스모니터링 기능을 제공하여 10초 간격으로 상대 노드의 상태를 점검하여 상대 릴레이 노드가 NOT_ALIVE 상태인 경우 전송을 중단하여 불필요한 트래픽 발생을 억제하는 역할을 수행한다.

3) 가상 종단포인트 관리자(VEM: Virtual Endpoint Manager)

원격 라우터로부터 EDP 정보(Subscriber)를 받을 경우 가상 Reader 등록 순서를 보여준다. 상대방의 라우터로부터 BuiltinSubscriber에 있는 cache를 전송을 받으면, 상대방의 PDP Guid를 먼저 검색을 하고, 그리고, 이를 바탕으로 상대 방의 cache에 있는 endpoint guid를 검색을 하고, Virtual Reader를 생성을 한다. 그리고, BuiltinPublisher의 History cache에 저장되어 있는 cache 정보를 비교하여, 일치되는 publisher정보를 연결시킨다. 반대로, 새로운 Publisher 데이 터가 전송이 되면, Virtual Reader들과 비교하여, 일치되는 Publisher를 연결시킨다.

4) 글로칼 토픽 관리자(GTM: Glocal Topic Manager)

릴레이 노드에서 사용되는 토픽은 다른 네트워크 도메인 과 데이터를 공유하는 Global Topic과 Local에서만 사용되 는 Local 토픽으로 구분된다. 릴레이 노드는 DDS도메인의 노드들이 보내는 EDP 메시지에서 토픽 정보를 수집하여 Global과 Local 토픽으로 구분하여 저장한다. Global/Local로 구분된 Topic은 가상 Endpoint 생성에 참조하게 된다.

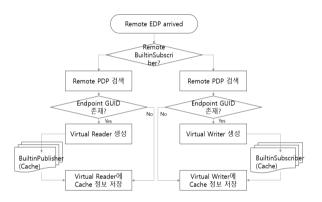


Fig. 11. VEM Processing Operation Flow

5. SIA-LVC 아키텍쳐 실험 및 분석

5.1 SIA-LVC 전송 프로토콜 검증

본 연구에서 SIA-LVC 전송 프로토콜로 자체 개발한 DDS (이하 EDDS)를 활용하며, 전송지연, 패킷전송율 및 CPU 점 유율 등의 성능 검증을 위해 Table 2 규격을 가진 시스템들 을 적용한다.

Table 2. Performance Test Environment of SIA-LVC Transport Protocol

| 노트 | 규격 | | | |
|--------------------|--|--|--|--|
| | - Intel(R) Corei7-3517 1.90GHz 2.40GHz | | | |
| Publisher | - RAM: 8GB | | | |
| | - OS: Win7 x64 | | | |
| | - Intel(R) Corei7-4500 1.80GHz 1.80GHz | | | |
| Subscriber | - RAM : 7.72GB | | | |
| | - Win7 x64 | | | |
| Network | 100Mbps, 1Gbps Ethernet | | | |
| 비교대상 상용 | DTI COMMENT DDC | | | |
| DDS | RTI CONNEXT DDS | | | |
| 전송 데이터 | 100Byte/패킷 | | | |
| 사이즈 | | | | |
| 0-0 | Reliability=RELIABLE/History=KEEP_AL | | | |
| QoS | L, Resource limit=max | | | |
| History Cache size | 5000 sample | | | |

Fig. 12에서 Fig. 14까지 SIA-LVC 전송 프로토콜의 성능 을 기존 DDS 상용 솔루션이며 성능상 세계 1위인 RTI사의 Connext DDS와 비교하였다. 패킷전송율은 EDDS가 25%이 상의 전송속도를 제공하는 반면, 전송지연시간은 평균 50% 높고 CPU 처리부하는 RTI사가 매우 높은 최적화 수준을 보이고 있다.

다만, 전송지연시간은 수신 노드의 수가 증가할수록 EDDS는 일정하게 유지되나 Connext DDS는 지연시간이 크게 증가하고 있어, EDDS가 패킷전송율과 함께 확장성면 에서 좋은 특성을 보이는 것을 알 수 있다. 이는 대규모 L-V-C 훈련체계 구축에 긍정적 효과를 보일 것으로 사료 된다.

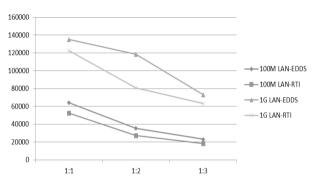


Fig. 12. The Comparison of Packet Transmission Rate (packet/sec)

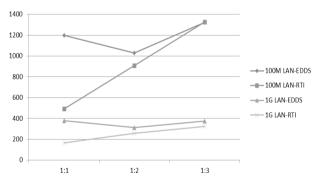


Fig. 13. The Comparison of Packet Transfer Delay (usec)

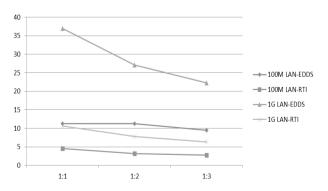


Fig. 14. The Comparison of CPU Usage

5.2 SIA-LVC 게이트웨이 검증

SIA-LVC 게이트웨이의 핵심 기능인 이종 미들웨어 API 연동을 위한 변환 기능으로 IEEE HLA와 OMG DDS 간의 연동을 실험한다. 특히, HLA의 132개 RTI API와 49개 콜백 API, DDS의 233개 API를 분석하여 결과적으로 본 연구에서 연동시킨 HLA와 DDS간 표준 API 집합을 정리하면 Table 3과 같다.

Table 3, Standard API Conversion Table Between HLA and DDS

| Service | HLA | DDS | |
|---------------------------------------|--|--|--|
| | joinFederation Execution() | create_participant() | |
| FM (Federation Management) | resignFederation Execution() | delete_participant() | |
| | requestPause() | suspend_publication() | |
| | requestResume() | resume_publication() | |
| | publishObjectClass() subscribeObjectClass Attribute() | create_topic() create_datawriter() create_datareader() | |
| DM (Declaration Manag ement) | unsubscribeObject Class() unpublishObjectClass() | delete_datareader() delete_datawriter() delete_topic() | |
| | publishInteraction Class() subscribeInteraction Class() | create_topic() create_datawriter() create_datareader() | |
| | unsubscribeInteraction Class() unpublishInteraction Class() | delete_datareader() delete_datawriter() delete_topic() | |
| | registerObjectInstance() | register_instance() write() write_w_timestamp() | |
| OM | deleteObjectInstance() | dispose() dispose_w_timestamp() unregister_instance() | |
| (Object Management) | requestClassAttribute ValueUpdate() requestObjectAttribute ValueUpdate() updateAttributeValues() | write() write_w_timestamp() | |
| | sendInteraction() | write() write_w_timestamp() | |
| | requestClassAttribute ValueUpdateWith Region() | create_participant() | |
| DDM (Data Distribution Man | associateRegionFor Updates() subscribeObjectClass AttributesWithRegion() | delete_participant() | |
| agement) | sendInteractionWith Region() | suspend_publication() | |
| | registerObjectInstance WithRegion() | resume_publication() | |

HLA/RTI와 DDS를 SIA-LVC 게이트웨이를 통해 실행한 글로벌 페더레이션 참가를 위한 미들웨어 연동 결과를 Fig. 15에 도시한다. 이 결과는 HLA 도메인에서 글로벌 페더레이션에 가입하는 동작으로 결과적으로 GW 내에 DDS 도메인과 통신할 수 있는 datawriter와 datareader 및 토픽이 생성시킨다. 이후 DDS 도메인에서 participant들이 동일한 글로벌 패더레이션에 참여하기 위해서는 각각 datawriter, datareader 및 토픽을 생성하고 해당 토픽에 pub/sub하면데이터 연동이 이루어진다.

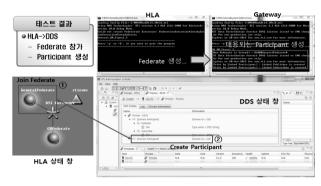


Fig. 15. HLA/RTI Initiating Global Federation Join Operation

최종적인 HLA와 DDS간 데이터 연동결과를 Fig. 16에 도시한다. 이 실험결과를 통해 HLA/RTI 객체가 발생시킨데이터를 SIA-LVC 게이트웨이를 통해 DDS 객체가 수신하는 것을 검증하였다. 이종 미들웨어간 연동 시 소요되는 온톨로지기반의 변환지연시간은 평균 15msec로 추후 보다 최적화하는 연구가 필요할 것으로 파악되었다.

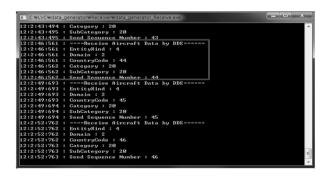


Fig. 16. DDS Object Receiving the Data Transmitted by HLA/RTI Federate

Fig. 17에서는 HLA/RTI의 시뮬레이션 모드 중 time-stepped 방식에 시간동기화 메카니즘을 적용한 결과이다. Time step이 커질수록 허용 가능 오차에 포함되는 메시지 개수는 증가하는 것을 알 수 있다. 메시지 개수가 1개/sec일 때, Time step에 관계없이 허용 가능 오차를 모두 만족한다. 그러나, 메시지 개수가 증가함에 따라 동기화의 수준이 낮아지고, 극단적으로 메시지 수가 1000개/sec일 때는 어떠한 Time step 인 경우에도 허용 가능 오차에 포함되지 않는다. Time step 관점에서 보면, Time step이 매우 짧은 1ms일 때는

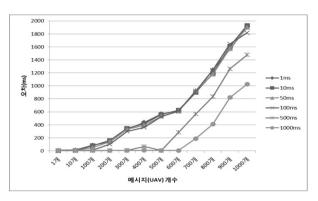


Fig. 17. Time-Stepped Mechanism: Time Synchronization Errors on Message Transmission Ratios

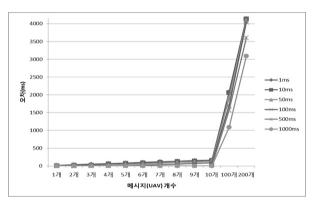


Fig. 18. Event-Driven Mechanism: Time Synchronization Errors on Message Transmission Ratios

메시지 개수가 1개/sec 일 때만 허용 가능 오차 (10msec)에 포함된다. 즉, 발생하는 메시지 수가 매우 제한되는 반면에 Time step이 1000ms일 때는 메시지 개수가 600개/sec 일 때에도 허용 가능 오차에 포함되어, 대규모 LVC 훈련체계설계 시에 정밀한 시간 동기화를 위해서는 HLA/RTI 도메인에서의 time step을 크게 정의하는 것이 유리함을 본 실험을 통해 도출하였다.

반면, event-driven 방식은 time-stepped 방식과 다르게 시간을 진행할 때 전송해야 될 메시지가 중간에 발생하게 되면 추가로 전송해야 되는 메시지 개수가 늘어난다. Fig. 18에서 도시하는 바와 같이, 메시지 개수가 10개/sec 이상에서 Time step 값에 상관없이 허용 가능 동기 오차를 초과하는 것을 알 수 있다. Event driven 방식은 Time stepped 방식과는 다르게 시간을 진행할 때 중간에 전송해야 될 메시지가 생기면 메시지 당 Time Grant, Time Advanced 메시지가 발생된다. 따라서 메시지량의 증가로 시간 동기화 오차는 급속도로 커진다. 또한, Time step이 큰 경우(1000msec)에도 작은 규모의 메시지 발생 (10개/sec)에도 동기 오차는 허용범위를 초과하게 된다.

따라서, 대규모 LVC 훈련체계에서의 글로벌 시간 동기화를 위해서는 event-driven 방식에 비해 time-stepped 방식이 유리하며, HLA/RTI 도메인에 가능한 큰 time step 사이즈를 설정하는 것이 적절하다는 것을 확인하였다.

5.3 SIA-LVC 릴레이 노드 검증

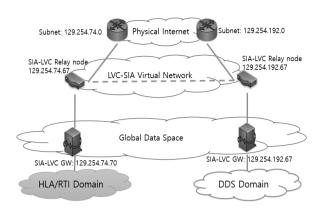


Fig. 19. SIA-LVC Virtual Network Topology on WAN

본 절에서는 릴레이 노드가 원격 도메인 릴레이 노드들의 DomainID와 IP 주소를 관리하는 라우팅 테이블 제공 기능, 로칼 도메인의 릴레이 노드와 원격 도메인 릴레이 노드 사이의 Discovery 기능 제공 및 원격 도메인의 참여자를 대신해 로칼 도메인 참여자에 대한 Discovery 기능을 제공하여 원격과 로칼 참여자간의 정상적인 테이터 전송 기능을 검증한다. 검증을 위한 네트워크 구조는 Fig. 19와 같다.

릴레이 노드 (129.254.192.67)에서 상대 릴레이 노드 (129.254.74.67)로 router_pdp()가 호출되어, 두 릴레이 노드 들간에 맺은 TCP 커넥션을 통해 라우팅 테이블이 전송된 과정을 Fig. 20에 도시한다. 이 절차를 통해 릴레이 노드에는 상대 도메인의 통신객체에 대한 가상 객체들이 자동 생성되어 로칼 객체들에게 원격 객체들이 마치 로칼 네트워크에 존재하는 객체로 인식하도록 한다. 결과적으로 SIA-LVC 가상 네트워크가 물리적인 인터넷 상에서 구축된다.

| | ***** | ************ | | |
|---|--|--|---------------------|---------------|
| 842 19.604401000 | 129.254.74.67 | 129.254.192.67 | TCP | 318 rtps-c |
| 843 19.0038/3000 | 129.234.192.0/ | 129.234.74.07 | TCP | 00 msua > |
| 844 19.605879000 | 129.254.192.67 | 129.254.74.67 | TCP | 62 m3ua > |
| 845 19, 605891000 | 129.254.74.67 | 129, 254, 192, 67 | TCP | 54 rtps-c |
| 846 19.607361000 | 129.254.192.67 | 129.254.74.67 | TCP | 318 m3ua > |
| 84/ 19.60/3/4000 | 129.254.74.67 | 129.254.192.6/ | TCP | 34 FTPS-0 |
| 871 19.907817000 | 129.254.74.67 | 129.254.192.67 | TCP | 66 11609 |
| 872 19.908598000 | 129.254.192.67 | 129.254.74.67 | TCP | 66 rtps-0 |
| 873 19.908625000 | 129.254.74.67 | 129.254.192.67 | TCP | 54 11609 |
| 874 19.910446000 | 129.254.192.67 | 129.254.74.67 | TCP | 62 rtps-0 |
| 875 19, 910459000 | 129.254.74.67 | 129.254.192.67 | TCP | 54 11609 |
| 876 19.911198000 | 129.254.192.67 | 129.254.74.67 | TCP | 318 rtps-0 |
| Internet Protocol Ve Transmission Control Source port: rtps- Destination port: | ersion 4, Src: 129 Protocol, Src Pordiscovery (7400) | .254.74.67 (129.254 | .74.67), | |
| ☐ Internet Protocol Ve Transmission Control Source port: rtps- Destination port: [Stream index: 4] Sequence number: 9 | ersion 4, Src: 129 Protocol, Src Podiscovery (7400) m3ua (2905) (relative seq | .254.74.67 (129.254 rt: rtps-discovery uence number) | (7400), | Dst: 129.254 |
| @ Internet Protocol vo # Transmission Control Source port: rtps- Destination port: [Stream index: 4] Sequence number: { [Next sequence num Acknowledgment num Header length: 20 | ersion 4, Src: 129 Protocol, Src Poidiscovery (7400) m3ua (2905) O (relative sequer: 273 (relative: 1 (relative) | .254.74.67 (129.254 rt: rtps-discovery | (7400), | Dst: 129.254 |
| ■ Internet Protocol V Transmission Control Source port: rtps- Destination port: [Stream index: 4] Sequence number: 6 [Next sequence num Acknowledgment num Acknowled | ersion 4, Src: 129 Protocol, Src Poudiscovery (7400) m3ua (2905) O (relative sequester: 273 (relative) bytes: 1 (relative) ACK) | .254.74.67 (129.254 rt: rtps-discovery uence number) tive sequence numbe | (7400), | Dst: 129.254 |
| E Internet Protocol W Transmission control Source port: rtps- Destination port: [Stream index: 4] Sequence number: 4; [Next sequence num Acknowledgment num Header length: 20 E Flags: 0x018 (PSM, Window size value: | ersion 4, Src: 129 Protocol, Src Poidiscovery (7400) m3ua (2905) (relative sequiber: 273 (relative sequiber: 1 (relative sequiber: 1 (relative sequiber: 1 (relative sequiber: 1 (relative sequiper: 1 | .254.74.67 (129.254 rt: rtps-discovery uence number) tive sequence numbe | (7400), | Dst: 129.254 |
| © Internet Protocol V □ Transmission control □ Source port: rtps: □ Destination port: □ [Stream index: 4] □ Sequence number: 5 □ [Next sequence number: 4 □ [Next sequence number: 5 □ [Next sequence number: 4 □ | ersion 4, Src: 129 I Protocol, Src Por discovery (7400) m3ua (2905) gua (2905) there: 273 (relative sequence: 1 (relative sequence: | .254.74.67 (129.254 rt: rtps-discovery uence number) tive sequence numbe | (7400), | Dst: 129.254. |
| © Internet Protocol W Transmission Control Source port: rtps: Destination port: [Stream index: 4] Sequence number: [Next sequence num Acknowledgment nu Acknowledgment nu Middow size value [Calculated window [Window size scal*] | ersion 4, Src: 129 Protocol, Src Pordiscovery (7400) maua (2905)) (relative sequester: 273 (relative: 1 (relative) bytes ACK) : 16425 s size: 65700] ing factor: 4] | .254.74.67 (129.254 nt: rtps-discovery uence number) tive sequence number ve ack number) | (7400), | Dst: 129.254. |
| in Internet Protocol V in Transmission Control Source port: rtps- Destination port: [Stream index: 4] Sequence number: 1 [Next sequence num Acknowledgment num Header length: 20 in Flags: 0x018 (Psh. window size value: [Calculated window [window size scal' in Checksum: 0xb761] in Checksum: 0xb761] | ersion 4, Src: 129 Protocol, Src Por discovery (7400) m3ua (2905)) (relative sequiber: 273 (relative) ber: 1 (relative) ACK): 16425 r size: 65700] ing factor: 4] validation disabl | .254.74.67 (129.254 nt: rtps-discovery uence number) tive sequence number ve ack number) | (7400), | Dst: 129.254 |
| ② Internet Protocol V □ Transmission Control Source port: rtps- Destination port: [Stream index: 4] Sequence number: 1 [Next sequence num Acknowledgment num Beader length: 20 □ Hander length: 20 □ Hander length: 20 □ Calculated window [Vindow size value: □ Checksum: 0xb761 □ [SEG/ACK analysis] | ersion 4, Src: 129 Protocol, Src Por discovery (7400) m3ua (2905)) (relative sequiber: 273 (relative) ber: 1 (relative) ACK): 16425 r size: 65700] ing factor: 4] validation disabl | .254.74.67 (129.254 nt: rtps-discovery uence number) tive sequence number ve ack number) | (7400), | Dst: 129.254 |
| © Internet Protocol V. □ Transmission Control Source port: rtps- Destination port: [Stream index: 4] Stream index: 4] Source port: [Stream index: 4] Source port: [Acknowledgment num Acknowledgment num Header length: 20 □ Flags: 0x018 (PSM. Window size value: [Calculated window [Window size value: □ [SEG/ACK analysis: □ Data (264 bytes) | rsion 4, Src: 129 Protocol, Src Pod discovery (7400) m3ua (2905)) (relative seq sber: 273 (relative bytes . AcK) .16425 rsize: 65700] ing factor: 4] validation disable | .254.74.67 (129.254 rt: rtps-discovery uence number) tive sequence number ve ack number) | (74.67), (7400), | Dst: 129.254 |
| © Internet Protocol V. □ Transmission Control Source port: rtps- Destination port: [Stream index: 4] Stream index: 4] Source port: [Stream index: 4] Source port: [Acknowledgment num Acknowledgment num Header length: 20 □ Flags: 0x018 (PSM. Window size value: [Calculated window [Window size value: □ [SEG/ACK analysis: □ Data (264 bytes) | rsion 4, Src: 129 Protocol, Src Pod discovery (7400) m3ua (2905)) (relative seq sber: 273 (relative bytes . AcK) .16425 rsize: 65700] ing factor: 4] validation disable | .254.74.67 (129.254 nt: rtps-discovery uence number) tive sequence number ve ack number) | (74.67), (7400), | Dst: 129.254. |

Fig. 20. SIA-LVC Virtual Network Construction
Based on Relay Nodes

릴레이노드에 의해 구축된 SIA-LVC 가상 네트워크 상에서 SIA-LVC 게이트웨이들간에는 개념적인 글로벌 데이터 공간이 형성되고, 로칼 미들웨어에서 DDS 포맷으로 변환된모든 데이터는 글로벌 데이터 공간에서 공유될 수 있다.

6. 결론 및 향후 연구

본 논문에서는 실제 훈련체계인 Live 체계, 무기체계 훈련용으로 활용되는 가상 훈련체계인 Virtual 체계 및 전략전술 훈련체계인 Constructive를 통합하여 적은 비용에 훈련효과도를 극대화하는 새로운 L-V-C 훈련체계 연동 기법인 SIA-LVC 아키텍쳐에 대한 연구결과를 상세히 소개하였다. 각 훈련체계들이 갖는 이종의 로칼 연동 미들웨어, 이종 데이터 포맷 및 세션 관리를 일관되게 연동하고, 글로벌 시간관리, 글로벌 데이터 공간 생성 및 관리 등에 대한 내용을다루었다.

SIA-LVC의 핵심요소들로는 SIA-LVC 전송 프로토콜, 게이트웨이 및 릴레이 노드가 있으며, 본 논문에서는 전송 프로토콜 측면에서는 세계적 경쟁 솔루션과 비교한 우수성과 대규모 L-V-C 훈련체계에 대한 적합성을 검증하였고, 게이트웨이 측면에서는 이종 미들웨어간 정밀한 API와 데이터 수준 비교와 분석을 통해 공통 기능에 대한 변환을 자동 수행할 수 있는 기반을 마련하였고, 글로벌 시간 동기화를 위한 time-stepped기법과 event-driven 기법을 실험하고 Live 체계와의 연동시 time-stepped 기법이 적합함을 검증하였다. 마지막으로, 릴레이 노드에 의한 물리적 인터넷 환경에서 SIA-LVC 가상 네트워크 구축 및 궁극적인 글로벌데이터 공간 구축을 통해 분산된 L-V-C 훈련체계들이 동일한 데이터 공간에서 정보를 공유할 수 있는 인프라를 구축하는데 성공하였다.

향후 연구과제로는 HLA/RTI의 출현 이전에 DIS 프로토콜이 Virtual 체계에 많이 활용되었고, 현재 운용 중인 전투기 시뮬레이터에 대부분 탑재되어 있어 본 논문에서 다룬 HLA/RTI와 DDS 간 연동 외에 DIS와의 연동에 대한 연구를 추진해야 할 것이며, 미국의 대표적인 L-V-C 연동 아키텍쳐이나 국내에는 적용된 바가 없는 TENA(Test and Training Environment Architecture)와의 연동이 필요할 것으로 사료된다.

References

- [1] Maneesh Varshney, Kent Pickett, Rajive Bagrodia, "A Live-Virtual-Constructive (LVC) framework for cyber operations test, evaluation and training," MILCOM 2011 Military Communications Conference, pp.1387-1392, 2011.
- [2] IEEE Computer Society, "IEEE Standard for Distributed Interactive Simulation - Application Protocols," SISO Strandards Activity Committee, New York, 2012.
- [3] IEEE Computer Society, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) Framework and Rules," Simulation Interoperability Standard Organization/Standards Activities Committee (SISO/SAC), New York, 2010.

- [4] F. Kuhl, R. M. Weatherly, J. Dahmann, Z. J. Fu, and Y. H. Wang, "Creating Computer Simulation Systems: An Introduction to High Level Architecture," National Defense Industry Press, 2003.
- [5] V. Fay-Wolfe, L. C. DiPippo, G. Cooper, and R. Johnson, "Real-time CORBA," *IEEE Transactions on Parallel and Distributed Systems*, Vol.11, Issue 10, pp.1073–1089, Oct., 2000.
- [6] J. Russell Noseworthy, "The Test and Training Enabling Architecture(TENA)," IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications, 2008.
- [7] Richard. M. Fujimoto, "Parallel and Distribution Simulation Systems," John Wiley & Sons, Inc., 1999.
- [8] OMG, "Data Distribution Service for Real-time Systems Version 1.2," Object Management Group, 2007.
- [9] Simulation Interoperability Standards Organization (SISO), "Standard for Guidance, Rationale, and Interoperability Modalities (GRIM) for the Real-time Platform Reference Federation Object Model (RPR FOM), Version 2.0," SISO-STD-001-2015, Aug., 2015.
- [10] A. Butler, "Live, Virtual, Constructive: A New Paradigm For Training," Aviation Week & Space Technology, Jul., 9, 2015.
- [11] D. Hudson and R. Hill, "The art and science of live, virtual, and constructive simulation for test and analysis," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, Vol.11, No.2, pp.77–89, 2014.



김 원 태

e-mail: wtkim@koreatech.ac.kr
1994년 한양대학교 전자공학과(학사)
1996년 한양대학교 전자공학과(석사)
2000년 한양대학교 전자공학과(박사)
2005년 ㈜로스틱테크놀로지 기술이사
2015년 한국전자통신연구원 CPS연구실
실장/책임연구원

2015년~현 재 한국기술교육대학교 컴퓨터공학부 조교수 관심분야: CPS, IoT, BCI



박 승 민

e-mail:minpark@etri.re.kr

1981년 울산대학교 전자공학과(학사)

1983년 홍익대학교 전자공학과(석사)

2001년 충남대학교 전자공학과(박사수료)

1984년 LG전자 연구원

1984년~현 재 한국전자통신연구원

임베디드SW연구부 책임연구원

관심분야: 임베디드SW, Cyber-Physical Systems, L-V-C 체계