

# HLA 기반 실시간 분산 M-SAM 시뮬레이션에서 RTI 성능 측정 및 분석

## (Performance Measurement and Analysis of RTI in the HLA-based Real-time Distributed M-SAM Simulation)

최 상 영 <sup>†</sup>      조 병 규 <sup>\*\*</sup>      이 길 섭 <sup>\*\*\*</sup>  
(Sang-Yeong Choi) (Byung-Kyu Cho) (Kil-Sup Lee)

**요 약** HLA는 시뮬레이션 표준 아키텍처로서 민수 및 국방 시뮬레이션 분야에서 관심이 높아지고 있다. 국내외에서 위게임과 같은 구조적 시뮬레이션에서는 HLA를 적용한 성공적인 개발사례를 찾아 볼 수 있다. 그러나 네트워크 시뮬레이터나 인터넷 게임과 같은 실시간 분산 시뮬레이션에서는 그 적용 사례가 아직까지 관찰되고 있지 않다. 그 이유는 실시간 분산 HLA 시뮬레이션에서는 네트워크 수준에서 RTI를 통한 메시지 전송시간이 컴퓨팅 능력, 시뮬레이션 노드 개수, 전송방식, 전송패킷의 크기에 영향을 받고, 응용 수준에서 메시지의 사건처리시간이 메시지 처리방식에 의존하여 실시간 제약조건을 설정하기가 어렵기 때문이다. 따라서 본 논문에서는 차후 M-SAM 시뮬레이터의 개발을 위하여 RTI의 실시간 제약 조건의 수준을 고찰하고자 한다. 이를 위하여 LAN 환경에서 6대의 PC상에 HLA 기반의 실험용 시뮬레이터를 개발하고 RTI의 성능을 측정하고 분석하였다. 본 연구의 결과로서 이전의 연구에서 제시되지 않았던 다수 노드 환경에서 실제 운용 시나리오와 부하에 따른 메시지 지연시간, RTI 부하 및 RTI 패킷전송 비율에 대한 정량적인 특성 값을 얻을 수 있었고, 이 결과는 향후 M-SAM 시뮬레이터 또는 유사 응용체계의 개발에서 표적 수, 전송주기 및 메시지 처리방식에 대한 설정지침으로 활용이 기대된다.

**키워드** : HLA, 실시간 분산 시뮬레이션, 성능 분석, 메시지 지연, RTI 부하, M-SAM

**Abstract** The HLA is the simulation architecture standard that the civilian and military simulation communities are deeply interested in. We can find various successful practices applying HLA to constructive simulations such as war games in domestics and overseas. However, any case of real-time distributed simulations has not been reported. The reason is that a message transmission period via RTI in a network layer varies according to computing power, simulation nodes, transmission types, and packet size; further a message processing time in an application layer depends on its processing methods, thus too difficult to set up real-time constraints for the enhancement of a real-time resolution. Hence, in this paper we have studied the real-time constraints of RTI for the development of the M-SAM simulator. Thus we have developed a HLA based pilot simulator using 6 PC's in LAN and then measured and analysed the performance of the RTI. As the results of our work, we could obtain the quantitative values for message delay, RTI overhead and RTI packet transmission ratio by a real operation scenario and loads, which are not shown in the previous works. We also expect that the results can be used as a guideline to set up the number of targets, transmission frequency, and message processing method in the development of the M-SAM simulator and similar applications.

**Key words** : HLA(High Level Architecture), Real-Time Distributed Simulation, Performance Analysis, Message Delay, Runtime Infrastructure overhead, M-SAM(Midium-range Surface-to-Air Missile)

· 본 연구는 2003년 국방과학연구소의 지원 하에 수행된 「대공유도무기 실시간 분산 시뮬레이션 기반 프로그램 개발」에 대한 위탁연구 결과임

† 비 회 원 : 국방대학교 무기체계과 교수

sychoi@kndu.ac.kr

\*\* 비 회 원 : 국방과학연구소 유도무기체계부

bgcho@naver.com

\*\*\* 종 신 회 원 : 국방대학교 전산정보학과 교수

gislee@kndu.ac.kr

논문접수 : 2004년 5월 19일

심사완료 : 2004년 11월 26일

### 1. 서론

오늘날 시뮬레이션 기술은 민수 분야에서는 게임산업, 원격의료, 원격교육, 협업 등의 분야에 사용될 전망이고, 국방 분야에서는 가상 교육훈련과 체계 획득분야에서 활발히 적용되고 있다[1,2]. 이러한 시뮬레이션 기술은 정보통신 네트워크를 통한 분산화, 향상된 전산장비의 처리속도를 기반으로 대량의 자료 및 그래픽 처리를 이용한 가상화 방향으로 발전되고 있다.

HLA(High Level Architecture)[3]는 네트워크 가상 환경에서 이기종의 컴퓨팅 환경과 시뮬레이션 응용체계를 공통 미들웨어 기반에서 상호간에 연동을 시키고, 나아가서 시뮬레이션 구성요소의 재사용성을 촉진시키기 위해서 미국 국방성의 지원 하에서 개발된 범용 시뮬레이션 아키텍처이다.

HLA는 규칙(Rules), 객체 모델 템플릿(OMT : Object Model Template), 인터페이스 명세(Interface Specification)로 구체적으로 정의되며, 2000년 9월에는 IEEE 1516 표준으로도 채택되었다. RTI(Runtime Infrastructure)는 HLA 인터페이스 명세를 구현하고 있으며, HLA에 부합되는 시뮬레이션 응용체계를 개발하기 위해서는 반드시 사용해야 하는 미들웨어이다.

국방 분야에서 시뮬레이션을 개발할 때에 RTI를 사용하여 HLA에 부합하도록 하고 있고, 상용분야에서도 게임 시뮬레이션, 교육용 시뮬레이터 등에 HLA를 적용하려는 시도가 이루어지고 있다. 위게임과 같은 구조적(Constructive) 시뮬레이션에서는 HLA를 적용하여 성공적으로 개발한 사례를 찾아 볼 수 있다. 그러나 시뮬레이터나 인터넷 게임과 같은 실시간 분산 시뮬레이션에서는 아직까지 관찰되고 있지 않다.

그 이유는 실시간 분산 HLA 시뮬레이션에서는 네트워크 수준에서 RTI를 통한 메시지 전송시간이 컴퓨팅 능력, 시뮬레이션 노드 개수, 전송방식, 전송패킷의 크기에 영향을 받고, 응용 수준에서 메시지의 사건처리시간이 메시지 처리방식에 의존하여 실시간 제약조건을 설정하기가 어렵기 때문이다. 특히, HLA를 적용하여 RTI를 사용할 경우에는 메시지 송수신처리 시간이 상대적으로 길어져서 실시간 제약조건을 만족시키지 못하는 경우가 발생할 개연성이 높다. 이 경우는 임무중요(mission critical) 시스템의 경우에는 중대한 시스템 오류를 발생하게 된다.

최근에 국내의 국방관련부서에서 M-SAM (Medium-range Surface to Air Missile) 시뮬레이터를 LAN 환경에서 HLA에 부합하도록 개발할 전망이다. 이는 시뮬레이션 관리통제기, 공중위협 모의기, 다기능 레이더, 교

전통제기, 발사대, 비행 유도탄으로 구성되어 방공교전을 실시간으로 모의하도록 되어 있다[4].

본 연구의 목적은 실제 M-SAM 시뮬레이터와 유사한 실험용 시뮬레이터를 개발하고, M-SAM 운용모드와 임무유형을 고려한 시나리오를 적용한 시뮬레이션을 통하여 RTI 노드사이의 메시지 지연시간, RTI 부하, 그리고 RTI 패킷전송 비율을 측정하고 분석하여 M-SAM 시뮬레이터 개발을 위한 실시간 제약조건을 예측하며 나아가서 설정지침을 제공하는 데에 있다.

현재 가용한 RTI에는 미 국방성의 DMSO(Defense Modeling and Simulation Office)에서 공개하고 있는 RTI와 상용화되어 있는 MÄK RTI와 Pitch RTI가 있다. 본 연구에서 사용한 RTI는 미 국방성의 DMSO에서 제공하고 있는 RTI 1.3NG-v3.2이다. 그리고 RTI 성능 측정 및 분석을 위해서 우리는 LAN으로 연결된 6대의 PC상에 RTI 기반의 6개 실험용 M-SAM 페드레이션 소프트웨어를 개발하여 사용하였다.

이후의 본 논문의 구성은 2장에서 관련 연구에 대하여 고찰하고 3장에서 RTI 성능측정 및 분석을 위한 실험용 페드레이션 소프트웨어 개발 및 실험환경 구축, 그리고 실험 시나리오에 대하여 설명하며, 4장에서는 실험조건과 방법에 대해서 설명한다. 5장에서는 실험결과를 분석하고, 마지막으로 6장에서는 주요 발견사항을 포함한 결론을 맺는다.

### 2. 관련 연구

#### 2.1 RTI 부하

RTI를 사용한 메시지 송수신 과정을 시뮬레이션의 시간 프레임(time frame)[5]에서 파악할 수 있는데, 시간 프레임 내에서 그림 1에서 보는 바와 같이 메시지 수신(tick), 사건처리(process events) 그리고 메시지 송신(send updates)이 이루어진다.

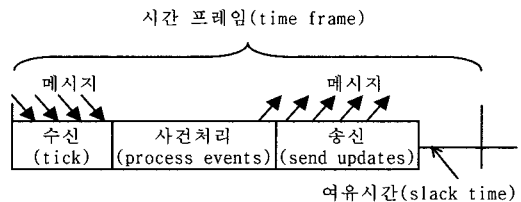


그림 1 시간 프레임

RTI를 사용할 경우에는 Tick()함수의 호출을 통하여 응용에서 메시지를 최종적으로 수신하게 된다. 이 과정에서 통신 인터페이스로부터 시뮬레이션 엔진의 각 채널로 수신되는 메시지들에 대하여 폴링과 LBTS(Lower

Boundary Time Stamp) 계산을 수행하고 이를 엔드 포인터인 로직 프로세스로 전달하여 스케줄링이 이루어진다. 이에 따라 메시지 수신 간에 RTI 폴링과 계산부하가 발생하고 이것이 물리적인 통신 부하와 더불어 메시지 지연의 주요 원인이 되며, 때로는 일정한 시간 프레임 초과함으로써 실시간 성능을 저하시킬 수 있다. 본 연구에서는 이러한 RTI 폴링 및 계산부하를 간단히 RTI 부하라고 하기로 한다.

전통적인 TCP/IP기반의 실시간 분산 시뮬레이션의 메시지 통신에서는 API 함수를 사용하여 메시지를 송수신 함으로써 이러한 부하는 거의 무시할 정도로 적었지만, RTI를 사용할 경우에는 다르다. 그래서 RTI 부하는 실시간 분산 시뮬레이션 설계에서 있어서 시뮬레이션의 시간 프레임을 결정하는 데에 중요하게 고려되어야 할 요소가 되고 있다.

## 2.2 RTI 성능실험 사례

실시간 분산 시뮬레이션은 DIS(Distributed Interactive Simulation)[6]에서 비롯된다. DIS에서는 프로토콜 기반으로 메시지를 송수신하고 Dead Reckon이라는 알고리즘을 통하여 메시지 처리시간을 동기화 시킨다. HLA에서는 RTI의 Tick() 호출을 통하여 메시지 송수신을 처리하고 보수적(Conservative) 알고리즘과 낙관적(Optimistic) 알고리즘을 사용하여 메시지 인과성 오류를 방지한다. HLA를 적용하여 실시간 분산 시뮬레이션을 수행할 때는 RTI에 의한 메시지 지연시간이 실시간 제약조건 만족여부에 영향을 주기 때문에 실시간 시뮬레이션 개발자에게 관심의 대상이 되어왔다.

Herbert[7]는 Windows 2000과 Linux 환경에서 두 개의 페드레이트 사이에서 시간 프레임을 40ms부터 400ms까지 변화시키면서 DMSO RTI NG1.3을 사용하여 RTI 성능을 실험하였는데 시간 프레임의 크기에 따라 메시지 손실량이 상당히 민감하게 나타나고, 300ms 정도 범위 내에서 메시지 손실이 없이 대부분이 수신되고 있다. RTI NG1.3의 서로 다른 버전에서도 유사한 결과를 보여주고 있다.

Darrell[8]은 RTI 테스트 보고서에서는 메시지 지연시간이 패킷 크기, 전송주기, 전송방식, 그리고 시뮬레이션 노드 수량에 따라 크게 영향을 받는다는 것을 보여주고, 노드 3개 정도까지, TCP 기반으로 전송할 경우에 메시지 크기를 1001개까지 증가시키면서 실험하였는데, 평균 메시지 지연시간이 100ms에서 300ms까지 나타나는 것으로 관찰되고 있다.

이 두 가지 대표적인 실험의 경우를 볼 때 메시지 지연시간이 실험환경과 조건에 따라 상당히 민감하게 나타나는 것을 알 수 있다. 또한 앞 절에서 언급된 바와 같이 메시지 지연시간의 원인이 TCP/IP 이계층층의 네

트워크 레벨에서 통신 지연시간도 있지만 오히려 그 상위계층인 미들웨어 레벨의 RTI 부하가 상당한 원인을 제공할 것으로 인식된다.

## 3. 실험용 페드레이션 개발 및 실험환경 구성

본 논문에서 관심이 있는 M-SAM 시뮬레이터의 경우는 기존에 Herbert와 Darrell이 수행한 실험환경과 상당히 다르다. 이들의 경우는 시뮬레이션 노드의 개수가 각각 2개와 3개인 반면에 본 실험의 경우는 6개이다. 또한 실험 시나리오도 다르다.

그래서 본 논문의 목적에 부합하는 실험환경을 구축하기 위해서 M-SAM 시뮬레이터와 유사한 실험용 페드레이션을 개발하여 사용하고 실험 시나리오도 현실과 유사하게 설정하였다. 또한 기존의 연구에서는 메시지 지연시간을 중심으로 측정했는데, 본 연구에서는 이 뿐만 아니라 순수한 RTI 부하와 RTI 패킷의 전송비율에 대해서도 상세히 측정하며, 응용수준에서도 메시지 처리 방식에 따라 메시지 지연시간의 변화량을 측정함으로써 보다 현실적인 결과를 얻고자 하였다.

### 3.1 실험용 페드레이션 개발 및 실험환경

실험환경을 구축하기 위하여 실험용 페드레이션을 개발하였는데, 페드레이션은 6개의 페드레이트 소프트웨어로 구성하고 C++로 개발하였다.[9] 여기서 6개의 페드레이트는 시뮬레이션 관리통제 페드레이트(SMCC: Simulation Management and Control Computer), 공중위협 모의 페드레이트(ATS: Air Threat Scenario), 다기능 레이더 페드레이트(MFR: Multi-Function Radar), 교전통제 페드레이트(ECS: Engagement Control System), 발사대 페드레이트(LAU: Launcher/Missile), 비행 유도탄 페드레이트(MSL: Missile/Flight)이다.

한편, 물리적 실험환경 구축을 위해서 6대의 PC(펜티엄 1.8GHz, Windows XP Pro)를 100Mbps 스위칭 허브를 이용하여 스타형 LAN으로 구성하였다. 그리고 6개의 페드레이트 소프트웨어를 각 PC에 탑재시켜 최종적으로 시뮬레이션 실험환경을 구축하였다.

각 페드레이트에서는 주요기능이 구현되고 페드레이트 사이에서 교환되는 메시지에 대해서는 현실과 유사하게 구현되었다.

### 3.2 실험 시나리오 설정

실험 시나리오는 M-SAM의 운용모드와 임무유형을 고려하여 그림 2에 나타난 바와 같이 이루어진다. 실험 시나리오에 따라 M-SAM 페드레이션 소프트웨어를 실행시켜 실시간으로 메시지가 송수신 되게 함으로써 실험측정치를 수집하고 분석하도록 하였다. 그림 2에서 화살표의 방향은 메시지의 흐름 방향을 나타낸다. 실선은 객체 메시지(object message) 흐름, 점선은 상호작용

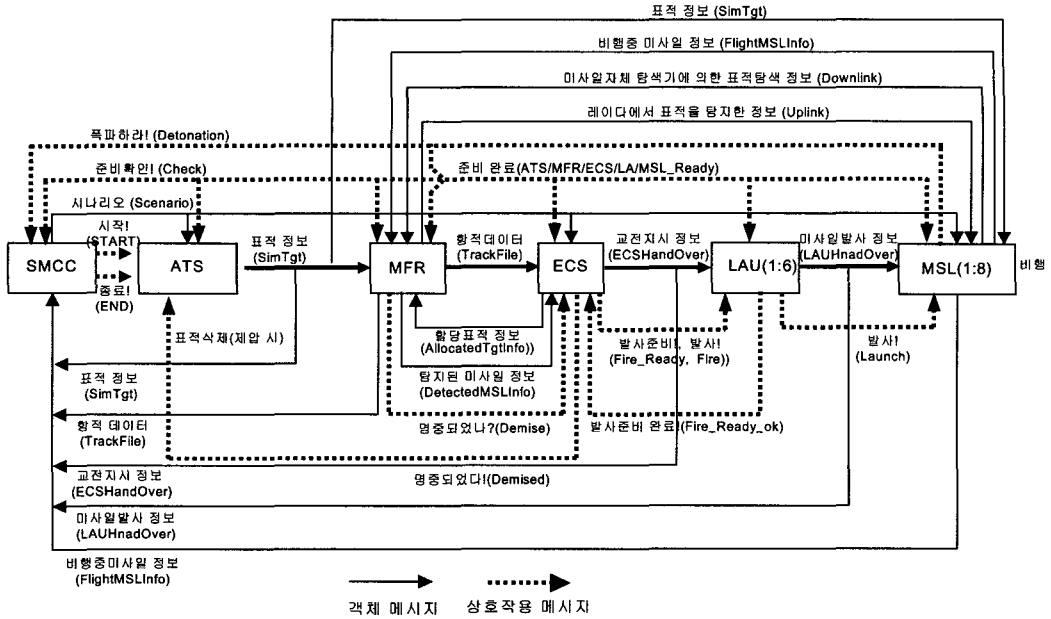


그림 2 실험 시나리오

메시지(interaction message) 흐름을 나타낸 것이다.

최초에, 시뮬레이션의 시작은 SMCC에서 ATS로 'Start' 상호작용 메시지를 송신함으로써 시작된다. ATS는 5개로부터 15개까지의 가상표적(SimTgt)을 생성하여 MFR과 MSL로 송신한다. MFR은 비행표적을 탐색하여 획득된 표적을 트랙 데이터(TrackFile)로 ECS로 송신한다. ECS는 획득된 트랙 데이터를 바탕으로 데이터 융합을 거쳐 위협평가와 표적할당을 실시하고 표적할당 정보(ECSHandOver)를 LAU와 MFR에 전송하고 동시에 발사준비(Fire\_Ready)를 지시한다. 발사대는 발사준비 완료(Fire\_Ready\_Ok)를 ECS에 통보한다.

이후 ECS는 교전(Fire) 명령을 하달한다. LAU는 명령을 하달 받아 미사일에게 발사지시 메시지(Launch)를 보낸다. ECS는 즉시 초기 비행정보(FiredMSLInfo)를 MFR로 송신하여 비행 MSL을 Lock-on하여 추적하도록 한다. MFR은 MSL에게 정보(Uplink)를 지속적으로 전송한다. MSL은 비행표적에 접근하면 폭파하고, ECS는 피해 평가를 실시하게 된다. 평가 후에 제압되었다면 제압 되었다는 정보(Demised)를 전송하여 제압된 표적에 대해서는 더 이상 생성되지 않도록 ATS에 전달한다. 그리고 각 페드레이트의 상태정보를 SMCC로 전송하게 된다.

실험 시나리오 상에서 페드레이트 간에 교환되는 데이터를 요약하면 표 1과 같다. 이는 객체(object)와 상호작용(interaction) 메시지로 구분되고 실제 구현상에서

표 1 시뮬레이션 교환 정보

객체(object) 메시지: 2732 byte		상호작용(interaction) 메시지: 128 byte	
객체 종류	데이터 크기	상호작용 종류	데이터 크기
Scenario	120 byte	CHECK	8 byte
SimTgt	88 byte	START	8 byte
Track File	88 byte	ATS_READY	8 byte
ECSHandOver	1024 byte	MFR_READY	8 byte
LAUHandOver	1024 byte	ECS_READY	8 byte
AllocatedTgtInfo	24 byte	LAU_READY	8 byte
FlightMSLInfo	156 byte	MSL_READY	8 byte
Uplink	88 byte	FIRE_READY	8 byte
Downlink	32 byte	FIRE_READY_OK	8 byte
DetectedMSLInfo	88 byte	FIRE	8 byte
		LAUNCH	8 byte
		DEMISE	8 byte
		DEMISED	8 byte
		DETONATION	8 byte
		EXPLOSION	8 byte
		END	8 byte

는 FOM(Federation Object Model)으로 구현되었다.

#### 4. 실험조건 및 방법

##### 4.1 실험변수 구분

실험변수는 4가지를 고려하였다. 표적 수, 전송주기, 전송방식, 메시지 처리방식이다. 각 실험변수에 대하여 표 2에서 보는 바와 같이 표적 수의 경우는 3가지, 그리고 전송주기, 전송방식, 메시지 처리방식에 대해서는 각각 2가지씩 고려하였다.

표 2 실험변수 구분

표적 수	전송 주기	전송 방식	메시지 처리방식
5개 10개 15개	5Hz	UDP	RO
	10Hz	TCP	TSO

여기서, 표적 수는 전송되는 표적 개수를 의미한다. 표적 개수만큼 전송량이 달라지는데, 예를 들어서 표적 개수가 5개인 경우에는 5 \* 88 바이트만큼 전송된다. 여기서 88바이트는 하나의 가상표적에 대한 위치정보, 속도정보 등을 포함하고 있다. 이는 표 1의 시뮬레이션 표적 SimTgt의 패킷량이다.

전송주기는 그림 2의 실험 시나리오 상에서 ATS에서 표적을 MFR과 MSJ로 송신할 때의 전송주기를 의미한다. 전송주기가 5Hz라고 하는 것은 200ms마다 표적 개수만큼 전송량을 송신한다는 것을 의미한다. 표 1의 다른 메시지들은 사건발생 시점에서 비주기적으로 송신된다.

전송방식은 UDP(User Datagram Protocol)를 기반으로 하는 베스트 에포트(Best Effort) 방식과 TCP(Transmission Control Protocol)를 기반으로 하는 신뢰적(Reliable) 방식으로 구분된다. RTI가 근본적으로 이들 프로토콜 기반으로 구현되어 있기 때문에 통신방식에 따라서도 전송속도에 상당히 영향을 미칠 수 있기 때문에 조건에 포함하였다.

메시지 처리방식은 HLA에서 두 가지로 구분된다. 하나는 RO(Receiver Order)방식이고 다른 하나는 TSO(Time Stamped Order)방식이다. 전자는 송신 측에서 메시지에 시간정보를 포함하지 않고, 수신 측에서도 RTI의 시간 서비스를 사용하지 않으면서 메시지가 들어오는 순서대로 처리하도록 하는 것이다. 이런 방식에서는 RTI 부하가 다소 작지만 인과성 오류가 발생할 수 있다. 후자는 시간정보를 포함하여 시간 서비스를 사용하여 보수적 알고리즘으로 메시지의 인과성 오류를 방지하면서 수신 처리하는 것이다.

#### 4.2 실험조건

실험조건은 표 2를 근거로 표적 수를 기준으로 3가지 경우를 설정하였다. 표적이 5개인 경우에는 “경우 1”, 10개는 “경우 2”, 15개는 “경우 3”이다. 그리고 각 경우에 대하여 표 3에서 보는 바와 같이 전송주기, 전송방식, 그리고 메시지 처리방식에 대한 조건을 서로 달리하면서 총 8가지의 서로 다른 조건에서 시뮬레이션을 수행하였다.

#### 4.3 시뮬레이션 시간 동기화 및 측정 방법

실험 목적상 페드레이트들의 지역 시뮬레이션 시간을

표 3 실험조건 구분

조건	내용
5Hz-UDP RO	전송주기: 5Hz, 전송방식:UDP 메시지 처리방식: RO방식
10Hz-UDP RO	전송주기: 10Hz, 전송방식:UDP 메시지 처리방식: RO방식
5Hz TCP RO	전송주기: 5Hz, 전송방식: TCP 메시지 처리방식: RO방식
10Hz TCP RO	전송주기: 10Hz, 전송방식: TCP 메시지 처리방식: RO방식
5Hz-UDP TSO	전송주기 : 5Hz, 전송방식: UDP 메시지 처리방식: TSO방식
10Hz UDP TSO	전송주기: 10Hz, 전송방식: UDP 메시지 처리방식: TSO방식
5Hz TCP TSO	전송주기: 5Hz, 전송방식: TCP 메시지 처리방식: TSO방식
10Hz TCP TSO	전송주기: 10Hz, 전송방식: TCP 메시지 처리방식: TSO방식

동기화 시켜서 메시지의 송수신 시간을 측정하였다. 시간 동기화 방법은 NTP(Network Time Protocol) 기반의 타임 서버용 Tardis 2000과 클라이언트용 Symtime을 설치하여 타임서버에서 방송되는 시간신호를 참조하여 동기화 하였다. 그리고 시스템 타이머는 윈도즈 타이머를 사용했으며 이는 해상도가 10ms 정도인 것으로 알려져 있다.

메시지 지연시간과 RTI 부하를 측정하는 데 있어서 시스템 타이머를 이용하였다. 먼저, 메시지 지연 시간의 경우에는 송신 페드레이트에서 메시지를 보낼 때 시스템 타이머의 시간을 참조하여 시간정보를 붙여서 보내게 된다. 이어서 수신 페드레이트 쪽의 LRC(Local RTI Component)로부터 수신 페드레이트 FederateAmbassador의 역호출(Call back) 함수의 호출을 통하여 메시지를 넘겨받을 때의 시간을 기록한다. 여기에서 역호출 함수는 객체인 경우의 Reflect, 상호작용인 경우의 Receive 함수에 해당한다. 그 후에 송신 메시지에 포함된 송신 시각을 확인하고 수신 시각과의 차이를 계산하는 방식으로 데이터를 수집하여 평균으로 계산하였다.

RTI 부하를 측정하는 데에 있어서는 먼저 수신 페드레이트에서 Tick()함수를 호출하는 시점에서 시스템 타이머 시간을 참조하여 기록하고, Tick()이 완료되는 시점에서 다시 시스템 타이머 시간을 참조하여 기록함으로써 그 시간의 차이 값을 수집하여 평균하여 계산하였다.

한편, RTI 패킷 전송비율을 측정하는 데에 있어서는 시뮬레이션 진행 중에 Ethernet Card를 통해 처리된 Packet정보를 Ethereal이라는 소프트웨어를 사용하여 TCP/UDP 패킷량, 송수메시지 패킷량, 그리고 RTI 패킷량을 구분하여 측정하고 총 전송 패킷량에 대한 RTI

패킷량의 상대적인 비율로 계산하였다.

### 5. 성능실험 결과

#### 5.1 메시지 지연시간

그림 3과 그림 4는 실험조건에서 설명한 바와 같이 3가지의 실험 경우 표적의 크기가 각각 5개, 10개, 15개에 대하여 표 3의 8가지 실험조건 따라 실험환경의 6개 시뮬레이션 노드 중에서 2개 구간에 대하여 측정된 메시지 지연시간을 보여주고 있다. 그림 3은 공중위험 모의 페드레이트(ATS)와 다기능 레이더 페드레이트(MFR) 구간에서 측정된 메시지 지연시간이고, 그림 4는 다기능 레이더 페드레이트(MFR)와 교전통제 페드레이트(ECS) 구간에서 측정된 메시지 지연시간이다. 여기서 X-축은 3가지 실험 경우 즉, 표적 수 5, 10, 15 개를 나타내며, Y-축은 1/1000초 (ms) 단위의 메시지 지연시간을 나타낸다. 그래프는 표 3의 8가지 실험조건별로 시뮬레이션을 통하여 얻어진 결과들이다.

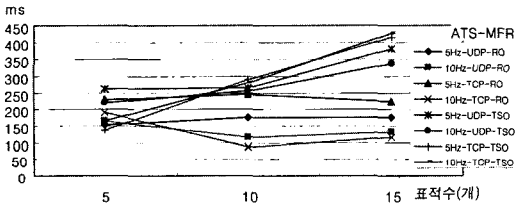


그림 3 ATS-MFR 구간의 메시지 지연시간

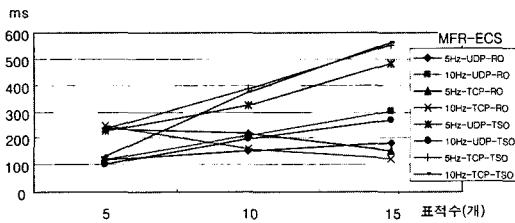


그림 4 MFR-ECS 구간의 메시지 지연시간

본 실험결과에서는 표적 수가 10개 이내의 경우(실험 2)는 최소 100ms에서 최대 400ms정도로 관찰되는데 이는 기존의 Herbert와 Darrell이 수행했던 실험결과와 유사한 결과를 주고 있다.

그러나 15개인 경우(실험 3)에는 기존의 결과와 상당한 차이를 보이고 있다. 일부 구간을 제외하고는 대부분 메시지 지연시간이 증가되면서 최고 600ms정도까지 발생하고 있다. 또한 전반적으로 메시지 지연시간이 길게 나타나고 있다. 이는 표 4에서 보는 바와 같이 기존의 실험환경, 노드수, 가변요소가 서로 다르기 때문이다.

표 4 기존 실험과 본 실험 결과 비교

구분	Herbert[7]	Darrell[8]	본 실험
실험 환경	LAN PC Windows 2000, Linux 2GHz	WAN IRIX SGI O2 180/200MHz	LAN PC Windows XP 1.8GHz
노드수	2 개	3 개	6 개
가변 요소	시간 프레임 (40~400ms)	메시지 크기 (~1001 Byte)	표적 수 (5개:440Byte~ 15개:1320Byte)
결과 요소	메시지 손실량, 지연시간	메시지 지연시간	메시지 지연시간, RTI 부하, RTI 패킷비율
주요 관찰 결과	300ms 이내 메시지 손실 없이 대부분 수신	메시지 지연시간: 100~300ms	표적 수 10개 이하에서 메시지지연시간 : 100~400ms
	-	-	표적 수 15개에서 메시지지연시간 : 100~600ms

실험결과에서 주목할 수 있는 것은 전송주기와 전송 방식에 따라서도 메시지 지연시간이 차이가 나는데, 기존의 실험에서 관찰되지 않았던 메시지 처리방식(TSO 혹은 RO)에 따라 상당히 차이가 나는 것을 발견할 수 있다. 특히, 전송량이 증가 할수록 더욱 많은 차이가 나는데, 표적이 15개인 경우는 거의 4배까지 관찰된다.

#### 5.2 RTI 부하

그림 5에서 그림 10까지는 3가지의 실험경우(표적의 크기가 각각 5개, 10개, 15개에 대하여 8가지 실험조건(표 3 참조)따라 본 실험환경에서 모든 시뮬레이션 노드 즉, 모든 페드레이트에서 측정된 RTI 부하를 보여주고 있다.

전반적으로 RTI 부하는 최소 5ms에서 최대 300ms까지 관찰되고, 각 페드레이트 공히 전송주기와 전송방식에 따라 차이가 난다. 또한 메시지 지연시간의 측정 결과에서 나타난 것과 마찬가지로 메시지 처리방식에 따라 현저한 차이를 보이고, 역시 전송량이 증가할수록 더욱 커짐을 알 수 있다.

TSO 방식의 경우에는 RTI 부하가 최소 100ms에서 최대 300ms 사이에 있고, RO 방식의 경우에는 40ms 이하로 관찰된다. 여기서 RO 방식이 TSO 방식보다 RTI 부하가 작은 이유는 RO 방식에서는 RTI 시간 서비스를 사용하지 않기 때문이다. 그리고 TSO 방식의 경우에는 RTI 부하가 상대적으로 크기 때문에 메시지 지연시간에 더 많은 영향을 끼칠 수 있다는 것이다.

이는 또한 임무건요 시스템이 아닌 단순한 그래픽이 강조되는 실시간 시뮬레이션에서는 TSO 방식 대신에

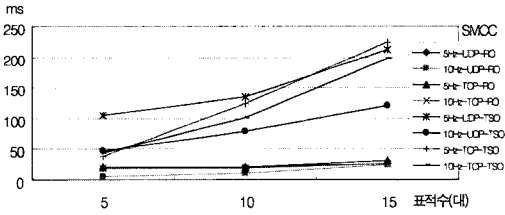


그림 5 SMCC 페드레이트에서 RTI 부하

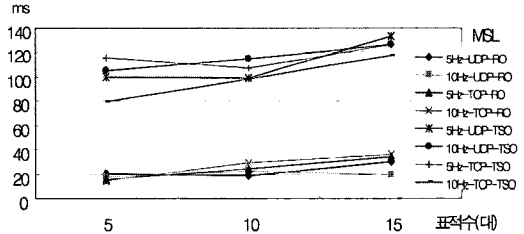


그림 10 MSL 페드레이트에서 RTI 부하

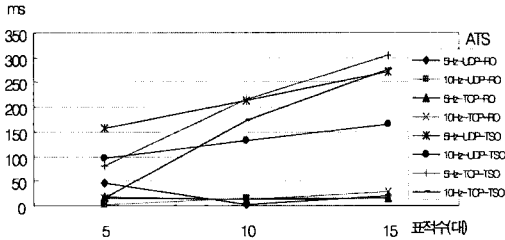


그림 6 ATS 페드레이트에서 RTI 부하

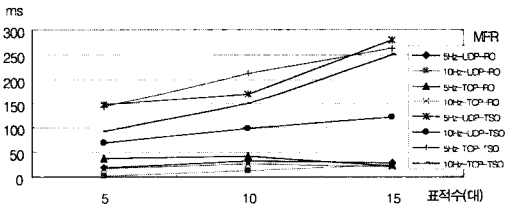


그림 7 MFR 페드레이트에서 RTI 부하

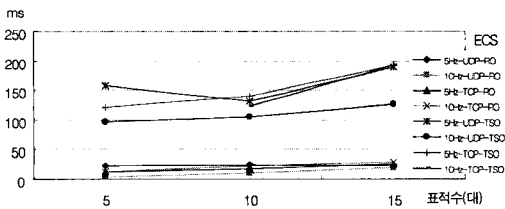


그림 8 ECS 페드레이트에서 RTI 부하

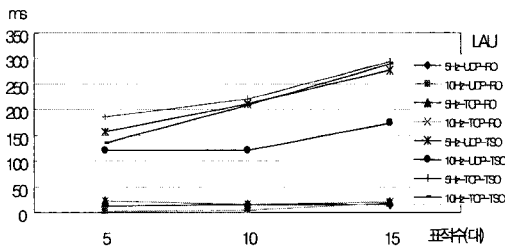


그림 9 LAU 페드레이트에서 RTI 부하

RO 방식으로 구현함으로써 RTI 부하 40ms정도를 고려하여 시간 프레임을 비교적 작게 설정하여 실시간 해상도를 높일 수 있다는 의미가 된다.

5.3 RTI 패킷전송 비율

그림 11은 시뮬레이션 진행 중에 Ethernet Card를

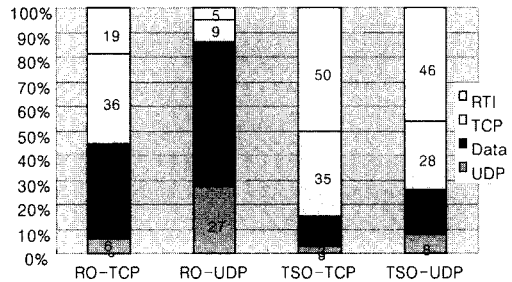


그림 11 RTI 패킷전송 비율

통해 처리된 TCP/UDP 패킷량, 송수메시지 패킷량, 그리고 RTI 패킷량에 대한 상대적 비율을 보여주고 있다. 그림 11에서 보듯이 전송방식과 메시지 처리방식에 따라 RTI 패킷전송 비율이 5%에서 50%까지 변화되고 있다. 특히, TSO 형태의 메시지 처리방식에서는 RTI 패킷량이 TCP/UDP 패킷량보다 더 많은 것을 알 수 있다. 이는 RTI 부하가 실시간 성능에 상당한 영향을 주고 있다는 것을 보여주는 것이다.

6. 결론

우리는 실험결과를 통하여 기존의 실험과 유사한 환경에서는 유사한 결과를 관찰할 수 있었다. 그러나 논문의 목적에 부합하도록 실험조건을 현실과 유사하게 엄격하게 부여한 경우에는 기존의 결과와 많은 차이를 보여주었다. 그리고 추가적으로 RTI 자체의 부하와 RTI 패킷전송비율에 대하여 세밀한 분석이 추가적으로 시도 되었으며 차후 개발될 M-SAM 시뮬레이터의 실시간 조건설정을 위한 다음과 같은 사실을 확인할 수 있었다.

첫째, M-SAM 시뮬레이터의 공중모의 위험기에서 표적 수를 5개에서 15까지 생성하여 모의할 때는 메시지 지연시간은 최소 100ms에서 최대 600ms까지 예측되며, 여기서 순수한 RTI 부하는 300ms까지 예측된다. 그래서 표적 생성을 최대 15대까지 고려할 경우에는 시뮬레이션 시간 프레임을 600ms 이상 설정해야 한다.

두 번째, HLA 실시간 분산 시뮬레이션에서 TSO방식의 메시지 처리방식이 RO 방식의 메시지 처리방식보다도 메시지 지연시간에서는 거의 2배, RTI 부하에서는

작게는 5배 크게는 10배 정도의 차이를 보였다. 그래서 임무근요 시스템이 아닌 소프트웨어 실시간 시뮬레이터에서는 RO 방식으로 구현함으로써 메시지 지연시간을 현격히 줄일 수 있다. 예를 들면 소규모 게임과 훈련용 시뮬레이터와 같이 그래픽이 강조되고 인과성 오류가 시스템 성능에 크게 영향을 미치지 않는 경우에는 RO 방식으로 구현함으로써 시간 프레임이 40ms 전후로도 가능할 것이다.

세 번째, 응용수준에서 RTI 서비스의 어떤 메시지 처리방식을 구현하는지에 따라 실시간 성능에 상당한 영향을 미치게 된다. 예를 들어서, RO 방식으로 구현할 경우에는 RTI 전송패킷 비율이 총 전송패킷량에 대하여 약 5%전후가 되지만 TSO 방식으로 구현할 경우에는 50%까지 감소해야 한다. RTI 전송패킷 비율이 높다는 것은 상대적으로 응용 메시지 패킷비율이 낮아짐을 의미한다.

이들 결과는 차후 M-SAM 시뮬레이터를 개발하는 경우에 실시간 제약조건에 대한 설정지침으로 제공될 수 있을 것으로 기대되며, 또한 본 연구의 결과가 HLA를 적용하여 소규모의 실시간 분산 시뮬레이션을 구현하고자 하는 경우에 RTI 성능 특성에 대한 세밀한 이해를 하는 데 도움이 되기를 기대한다.

### 참 고 문 헌

- [1] Piplani, L., Mercer, J. and Roop, O., Models and Simulations, Defense Systems Management College, 1994.
- [2] Kohl, F., Creating Computer Simulation Systems, Prentice Hall, 1999.
- [3] Defense Modeling and Simulation Office, High Level Architecture Interface Specification, Ver 1.3, available online at <http://www.dmsomil>, 1998.
- [4] 국방과학연구소, 대공유도무기 실시간 분산시뮬레이션 기반 프로그램 개발, MADC-416-030292, 2003.
- [5] McLean, T., Fujimoto, R., Middleware for Real-Time Distribution Simulation, available online at <http://www.gyu.gatech.edu/ccg/>, 2002.
- [6] Institute for Simulation and Training, The Standard for Distributed Interactive Simulation-Application Protocol, Univ. of Florida, Orlando, 1994.
- [7] Tietje, H., "Benchmarking of RTIs for Real-Time Applications," in Proc. of Simulation Interoperability Workshop, Orlando, 2003.
- [8] Wright, D., JADS JT&E High Level Architecture Runtime Infrastructure Test Report, JADS JT&E, New Mexico, 1998.
- [9] 국방과학연구소, M-SAM M&S 개발현황 및 추진방안, MADC-516-021020, 2002.



최 상 영

1982년 육군사관학교 이학사. 1985년 국방대학교 국방과학 석사. 1989년 (영)크랜필드 공대 운용/체계과학 박사. 2003년~현재 국방대학교 무기체계과 교수. 관심분야는 실시간 분산 시뮬레이션, 아키텍처 개발 방법론, 소프트웨어/체계 공학



조 병 규

1988년 인하대학교 전산학 학사. 1990년 인하대학교 전산학 석사. 2003년~현재 충남대학교 컴퓨터과학과 박사과정. 1990~현재 국방연 유도무기 체계부. 관심분야는 실시간 분산 시뮬레이션, HLA, 소프트웨어 공학



이 길 섭

1985년 금오공과대학 전자공학 학사. 1987년 한국과학기술원 전산학 석사. 1992년 한국과학기술원 전산학 박사. 2003년~현재 국방대학교 전산정보학과 순환직 교수. 관심분야는 정형기법, 분산 시스템 모델링, 소프트웨어 공학