

HLA/RTI 기반 항공교통관제 시뮬레이션에 효율적 객체 관리법

Object Management Method for Air Traffic Control Simulation Based on HLA/RTI

김명규*, 김정호*, 채수환**

Myung-Kyu Kim*, Jung-Ho Kim* and Soo-Hoan Chae**

요 약

HLA는 상호운용성과 재사용성을 목적으로 미국 국방성 산하 기관인 DMSO에서 개발되어진 분산 시스템 프레임워크이고 RTI는 HLA의 하부 구조로서 메시지 교환과 객체 관리 등의 통신 과정 중에 필요한 서비스를 제공한다. 그러나 HLA 기반 시뮬레이션은 규모가 커지거나 복잡해지면 통신 부하가 발생하여서 성능이 저하되는 문제점을 갖고 있다. 이러한 단점을 해결하기 위해 데이터 교환할 때, RTI와 TCP 소켓을 부분적으로 사용한 OMM으로 지속적인 송수신에 대한 통신 부하를 경감할 수 있는 방법을 만들었다. 이 방법을 교통관제 시뮬레이션 시스템에 적용했을 때, 유용함을 시뮬레이션을 통해 증명하였다.

Abstract

HLA is made for interoperability and reusability as distributed system framework under DMSO. RTI provides the necessary services for message transmission, object management as HLA 's understructure. But its performance can decrease as it becomes larger and more complex because of communication overhead. To solve this problem, RTI and TCP socket are partially used during data exchange and OMM is designed to efficiently manage objects. The efficiency of the proposed method is proved through simulation when it is applied to an air traffic control simulation based on HLA/RTI.

Key words : HLA(High Level Architecture), RTI(Run Time Infrastructre), OMM(Object Management Module)

I. 서 론

현재 분산 시뮬레이션 환경에서 HLA (High Level Architecture)는 다양하게 널리 사용되고 있다. 이것은 미국 국방성 산하 기관인 DMSO (Defense Modeling & Simulation Office)의 주도하에 개발되었는데, 분산

시뮬레이션의 표준화를 위한 개념적인 분산 시스템 프레임워크(framework)을 제공한다. 이를 통해 응용 프로그램 간에 상호 운용성(interoperability), 재사용성(reusability)을 제공함을 목적으로 하고 있다[8].

HLA 하부 구조로 RTI(Run Time Infrastructure)가 있는데, 이를 통해 데이터 교환 및 운용에 대한 조율,

* 한국항공대학교 컴퓨터공학과

** 한국항공대학교 항공전자정보통신공학부

· 제1저자 (First Author) : 김명규

· 투고일자 : 2009년 1월 6일

· 심사(수정)일자 : 2009년 1월 7일 (수정일자 : 2009년 2월 26일)

· 게재일자 : 2009년 4월 30일

Federate 인터페이스(Interface) 명세에 의하여 정의되어지는 서비스들을 제공한다[1].

초기 HLA는 JWAR(Joint Warfare System), JSIM(Joint Simulation System), JMASS(Joint Modeling and Simulation System) 등 군방 관련 시뮬레이터 개발에 주로 사용되었지만, 현재는 기능을 지속적으로 확대하여 일반 분산 시뮬레이터 개발에 적용할 수 있어 항만, 교통, 항공 시뮬레이션 분야에 적용되고 있다. 그러나 다양한 시뮬레이션 환경에 적용되도록 설계되었기 때문에 상호 운용성과 재사용성을 유지하기 위해서 시뮬레이션 자체가 점점 복잡해지게 되었다.

통합되어 규모가 커진 시뮬레이션에서는 객체의 속성 변화를 전달하기 위한 federate들 간의 데이터 교환 및 객체와 속성을 관리하기 위한 초기화 및 종료 절차에서 많은 통신 부하가 발생하고, 데이터 필터링을 위한 처리 시간도 증가하게 되었다. 통신부하 증가의 주된 원인은 시뮬레이션 수행 중, 객체의 변화되지 않는 속성 정보도 같이 전달하는 경우가 발생하기 때문이다. 기존 연구에서는 통신량을 감소방법으로 FOM(Federation Object Model)에서 불필요한 요소를 제거하는 방법을 제시하였다[2]. 본 논문에서는 이러한 문제를 해결하기 위해서 효율적인 객체 관리 모듈(OMM)을 항공교통관제 시뮬레이션에 적용하여 통신 부하의 변화에 대한 결과를 분석하고 평가하였다.

II. 관련연구

2-1 HLA를 이용한 시뮬레이션 구성

HLA는 여러 다른 환경의 시뮬레이션을 통합하여 큰 규모의 시뮬레이션이 가능할 수 있는 구조를 제공한다. 예를 들어 항공기 기종별 조종을 위한 시뮬레이션과 관제사들을 훈련하기 위한 항공교통관제 시뮬레이션을 HLA를 이용해 각각 통합한다면, 더 큰 규모의 항공 관련 시뮬레이션이 가능하다. [그림 1]에 보인 바와 같이 통합되어진 시뮬레이션을 federation이라 하고 federation에 참가하는 각 요소들

을 federate라 부른다. 시뮬레이션을 구현하기 위해서 필요한 요소는 federation과 federate 시뮬레이션 수행 중 지켜야하는 HLA 규칙, federate들 사이에서 정의된 인터페이스 명세, 시뮬레이션을 기술하기 위한 OMT(Object Model Template) 세 가지가 필요하다 [3]-[6].

2-2 RTI 서비스

RTI는 운영체제와 federate들을 사이에 놓여 있는 미들웨어(middleware) 소프트웨어이다. [그림 1]에서 보인 바와 같이 federation의 제어 관리, 각 federate들의 데이터 교환 등을 위하여 6가지 서비스를 지원한다[3].

- (1) Federation Management : federation 수행의 생성, 제어 관리, 수정, 삭제를 관리하는 서비스 집합
- (2) Declaration Management : 참가한 federate 들이 사용할 정보의 생성 및 전달을 관리하는 서비스 집합
- (3) Object Management : 참가한 federate들이 사용한 객체의 등록, 수정, 삭제 및 interaction의 교환을 관리하는 서비스 집합
- (4) Ownership Management : 참가한 federate들 사이의 속성에 대한 소유권 이동을 관리하는 서비스 집합
- (5) Time Management : federation 수행중의 시간진행을 관리하는 서비스의 집합
- (6) Data Distribution Management : 참가한 federate 들 통신구역을 정의하는 서비스의 집합

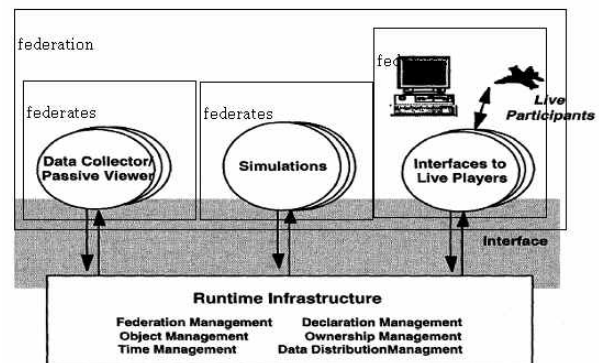


그림 1. HLA federation의 구조
Fig. 1. The structure of HLA federation

2-3 HLA 데이터 필터링 기법 [3]

RTI는 전송되는 데이터의 형식이나 내용물에 대해 상세히 알지 않아도 되는 구조를 가지며 단지 선언된 관심 영역에만 근거하여 데이터의 송신 측과 수신 측을 연결하게 된다.

분산된 환경에서 데이터를 효율적으로 전송하기 위하여 HLA는 DDM (Data Distribute Management)라는 데이터 필터링 기법을 제공한다.

DDM 서비스는 각 federate들의 관심 영역(region)을 위해 유연성 있고, 확장된 메커니즘을 제공하며 이 메커니즘은 RTI의 데이터교환 능력을 효율적으로 확장시켜 준다.

DDM에서 데이터의 효율적인 분배를 위하여 routing space를 선언한다. Routing space는 배열로 구성되어 있고, 각 배열은 각각의 영역을 정의한다. 영역은 범위(extent)의 집합으로 정의되며 각각의 범위는 가장 작은 형태의 영역 공간이다.

2-4 객체 관리 모듈 (Object Management Module)

HLA를 이용한 시뮬레이션에서 객체를 사용하기 위해서는 [그림 2]와 같은 절차를 수행해야 한다. 특

히, UpdateAttributeValues()를 통한 federate들 간의 속성의 값을 전달할 때, 불필요한 데이터가 발생하게 된다. [그림 3]과 같이 해당 객체의 속성들을 모두 묶은 구조AttributeHandleValueMap의 데이터 형을 사용하기 때문이다. 만약 해당 객체의 속성 값으로 x, y, z의 3차원 위치 좌표를 가진다면, 해당 객체가 오직 x 좌표만 이동하여도 좌표를 모두 전송해야한다. 이 경우 참여한 federate의 수가 많을 경우 발생하는 통신 부하는 큰 폭으로 증가한다. HLA에서 사용되어지는 통신 방법은 P2P(peer-to-peer) 방식이 아닌 브로드캐스팅(broadcasting) 방식을 사용하기 때문이다. 따라서 참여한 federate의 수가 많을 경우 발생하는 통신 부하는 배수로 증가한다.

이 문제를 해결하기 위해 OMM 모듈을 제안하였다. 이 모듈은 [그림 4]와 같이 구성되어 있다. HLA 코어(core)와 객체 관리 모듈 사이에 공통적으로 속성의 식별자(id)만을 유지하고 HLA 코어는 객체에 대한 정보를 포함하지 않는다. 즉, [그림 2]의 object class, object instance의 선언 부분이 HLA 코어에서 삭제되어진다. HLA 코어는 기존의 역할을 모두 수행하지만 객체 관리에 관한 권한만을 객체 관리 모듈에 이양한다.

기존 HLA의 구현에서 AttributeHandleValueMap을 사용하여 해당 객체의 속성을 일괄적으로 전송하던

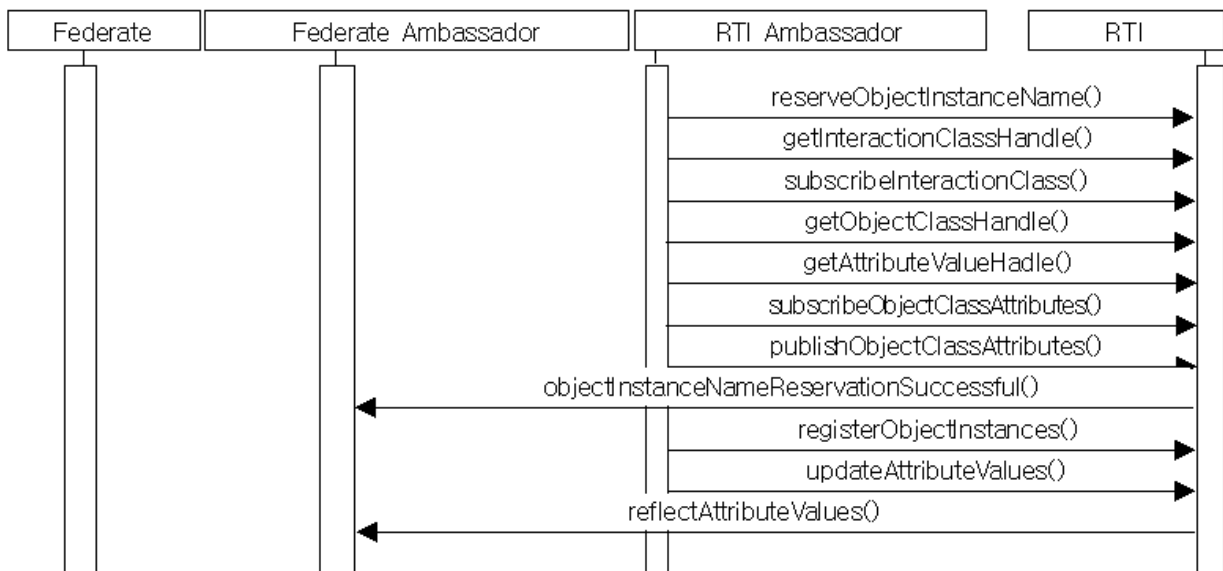


그림 2. HLA 객체의 정의 및 통신 순서

Fig. 2. The definition and sequence of communication on HLA's object

방식의 통신 부하를 줄이기 위해서 식별자(id)를 이용한다. interaction(id, 변화된 속성) 형태의 통신을 이용하여 변경된 속성만을 전달하게 된다[8].

hand e	-> add	(x_point	y_point	z_point)
	

그림 3. AttributeHandleValueMap의 구조
Fig. 3. The structure of AttributeHandleValueMap

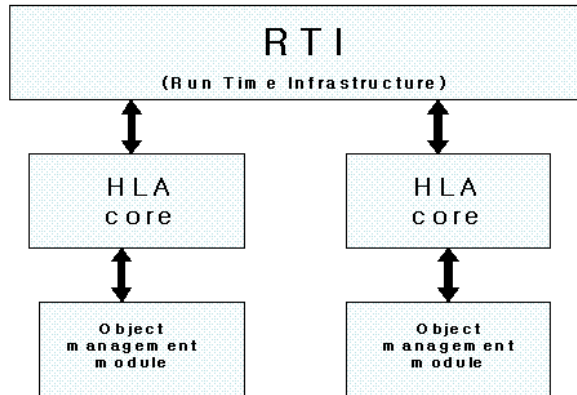


그림 4. OMM(객체 관리 모듈)의 구성
Fig. 4. The organization of OMM

2-5 HLA의 문제점

HLA 인터페이스 명세에 정의된 서비스들을 수행함에 있어서, RTI는 federation에서 이벤트 발생, 데이터 교환 등을 발생할 때, 전역 프로세스인 RTIExec와 FedExec가 반드시 존재해야만 한다. HLA의 첫 번째 단점이 바로 RTIExec와 FedExec의 존재이다. RTIExec는 각 federate들이 통신할 때 RTI 컴포넌트들을 초기화시키는 역할을 하고, FedExec는 각 federation을 관리한다. 시뮬레이션에서의 모든 federate와 federation은 RTIExec와 FedExec의 통제하에 있어야 한다. 예를 들면, RTI를 사용하는 모든 시뮬레이션에서 데이터 교환 시 발생하는 정보는 federation이 생성되기 전에 공표되어야 하고 federate들은 publishing, subscribing 데이터 정보를 공표해야만 한다. 이것이 바로 RTI에게 오버헤드를 발생시키

는 요인이다[9].

두 번째 단점으로는, RTI가 Tick() 메소드(method)를 호출한다는 것과 RTI는 멀티스레드(multi thread)를 제공하지 않는다는 것이다. RTI Tick() 메소드 호출이 RTI의 오버헤드를 가장 많이 발생시키는 요인이다. LRC (Local RTI Component)은 데이터를 교환하는 등 많은 일을 하고 이 작업을 수행하는데 시간이 필요하다. 이 때 Tick() 메소드는 RTI에게 시간을 제공하는 역할을 한다. 만약 Tick() 메소드를 호출하는데 실패한다면, federation에 참여하려는 일부 federate들이 참여할 수 없는 등 문제점들을 발생시킨다. 이 두 가지 단점 때문에 HLA의 규모가 커지고 복잡해지면 이로 인하여 federate 간 데이터 교환이 빈번하게 발생할 경우 HLA의 성능은 저하된다.

III. 시뮬레이션 모델링

HLA의 RTI 구현 형태는 중앙 집중 서버 모델과 분산 서버 모델로 구현할 수 있다. 중앙 집중 서버 모델은 전형적인 서버/클라이언트 모델로서 구조적으로 단순하여 구현하기 쉽지만, 확장성이 떨어지는 문제가 있다. 따라서 본 모델링에서는 시스템 확장성을 위해서 많은 하위 RTI들로 구성할 수 있는 분산 서버 모델을 채택하고 다만 시간 관리와 같은 전체적인 동작을 통제하는 호스트를 두었다.

시뮬레이션 모델의 성능을 측정하기 전에 시스템에서 RTI와 TCP 소켓(socket) 통신지연을 측정하였다. 이는 시스템 간 메시지를 송신하거나 수신하는 데이터 교환 능력과 관련이 있다. 본 시뮬레이션에서 하드웨어 사양은 펜티엄과 운영체제로 마이크로 소프트웨어사의 Window XP, RTI 버전으로 Pitch사의 pRTI 1516, 사용한 소켓의 종류는 마이크로소프트사의 윈속(winsock)을 사용하였다. 데이터의 용량을 1KB(Kilo Byte) 일 때의 왕복(round trip) 시간과 RTI의 왕복 시간을 10회 반복하여 구하였다.

또한 데이터 용량을 2KB, 4KB, 8KB, 16KB, 32KB로 증가시켜 데이터 크기에 따른 왕복 시간(round trip)을 비교하였다.

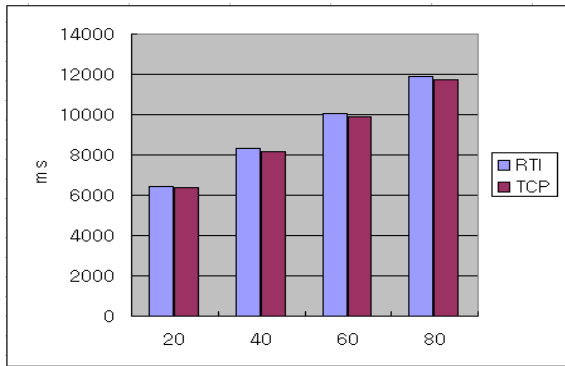


그림 5. 교환 횟수에 따른 시간
Fig. 5. Time by exchange counts

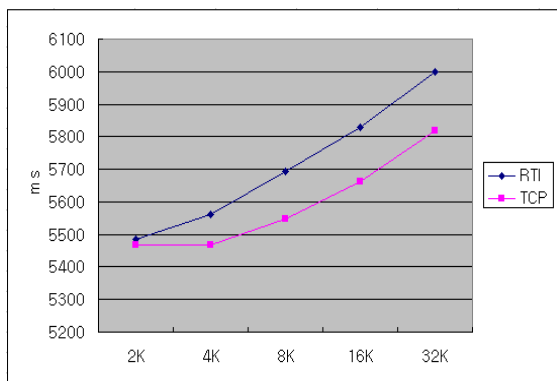


그림 6. 데이터양에 따른 시간
Fig. 6. Time by data's quantity

[그림5]에서 보는 바와 같이 TCP 소켓을 부분적으로 이용하였을 경우 HLA 시뮬레이션의 성능이 더 좋은 이유는 TCP소켓을 사용하였을 경우, RTIExec와 FedExec라는 전역변수와 Tick()메소드 호출 과정을 거치지 않기 때문이다.

[그림 5]와 [그림6] 결과에서와 같이 데이터 통신의 속도 면에서 살펴보면 TCP 만을 사용하는 것이 좋으나, 시뮬레이션의 유연성 확보를 위해서 RTI 기반의 분산 시뮬레이션 객체들을 관리하는 모듈을 적용하였다.

3-1 항공교통관제 시뮬레이션의 구성

본 연구에서는 OMM을 사용하여 항공교통관제 시뮬레이션을 구현한다. 기본적인 시뮬레이션의 구성은 [그림 7]에 보인 바와 같이 3개의 노드(3개의 federate로 구성)로 구성되어 있다.

Central RTI component는 전체 federation의 모니터

링을 수행한다. Federate 1, 2는 HLA core와 객체 관리 모듈을 포함하며 실질적인 시뮬레이션을 수행한다. 즉, 항공기에서 발생되어지는 정보를 이용하여 항공교통관제를 진행하는 각각의 공항에 해당한다.

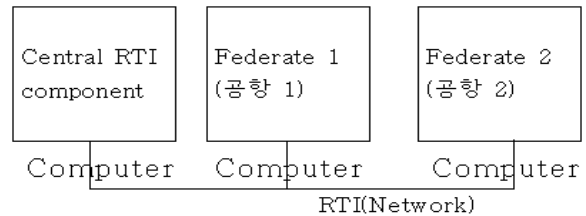


그림 7. 시뮬레이션 시스템 구성
Fig. 7. Organization of Simulation system.

Federate 1, 2의 객체 관리 모듈을 살펴보면 실질적인 시뮬레이션의 주체인 항공기 객체의 생성 및 파괴, 항공기 속성 값의 송수신을 포함한 정의를 하고 있다. 항공기 객체는 [그림 8]에 보인 바와 같이 모두 10개의 속성을 가지는 class Airplane으로 정의되어 있다.

```

class Airplane {
int    id;           //항공기 id
char   dep_airport; //출발 공항 정보
char   land_airport //도착 공항 정보
char   machine;     //항공기의 기체 정보
int    no_passenger; //탑승 승객 수
char   dep_time;    //출발 시간
char   land_time;   //도착 시간
float  point_x;     //항공기 좌표 정보 x
float  point_y;     //항공기 좌표 정보 y
float  point_z;     //항공기 좌표 정보 z
...
}
    
```

그림 8. Class Airplane의 구성
Fig. 8. Composition of Class Airplane

3-2 항공교통관제 시뮬레이션

시뮬레이션의 진행은 기본적으로 객체(항공기)의 생성, 비행, 도착 3단계로 수행된다. 객체의 생성 시, [그림 8]의 정보를 모두 포함한 비행기의 인스턴스(instance)가 만들어진다. 비행 이후 x, y, z의 좌표 정보는 수시로 변경되어진다. 하지만 이외의 7개의 정보들은 객체 생성 단계와 동일하다. 그렇지만 객체 관리 모듈을 적용하면 값의 변화가 있는 정보만을 전

송한다. 반면 제안한 OMM을 적용하지 않을 경우, HLA의 통신 특성상 모든 정보를 매 번 전송해야 한다. 각 단계에서 발생되어지는 통신에 포함되는 속성의 수는 <표 1>과 같다.

표 1. 통신에 포함되는 속성의 수
Table 1. Number of attribute in Communication

구분	객체 생성	비행	도착
OMM 적용	10개	3개	없음
OMM 미적용	10개	10개	없음

시뮬레이션에서 사용하는 객체의 수는 100개, 각 객체 생성 단계에서 1회, 비행 단계에서 4회의 통신이 federate간에 발생하도록 설정하였다.

IV. 시뮬레이션 결과 분석

객체 생성, 비행 단계에만 정보 전송이 이루어짐으로 이 두 단계에서 발생되어지는 통신 데이터 발생량을 각각 비교하고 시뮬레이션 종료까지의 전체적인 통신 데이터 발생량을 비교하여 OMM 적용 결과를 비교하였다.

객체 별로 객체 생성, 비행 각 단계에서 발생하는 통신 데이터 발생량과 객체 별 총 통신 데이터 발생량은 다음과 같다.

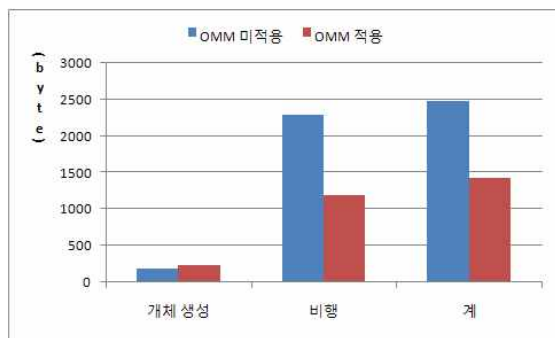


그림 9. 객체 별 생성시 통신 데이터 량
Fig. 9. Quantities of communication data by each object's creation

[그림 9]에 보인 바와 같이 객체 생성단계의 결과는 오히려 OMM을 적용한 경우 더 많은 통신 데이

터가 발생하였다. 객체 생성 시에는 1회 통신비용이 관리 모듈을 적용한 경우 더 많은 통신 데이터가 발생한다. 그 이유는 처음 객체 생성 시 10개의 모든 속성을 상대 federate에 전달해야하기 때문에 통신회수가 증가한다. 그렇지만, 비행단계에서는 변화되는 속성은 x, y, z 위치 속성 3가지만 변화하기 때문에 전달해야 하는 통신 데이터 량은 급격히 감소한다. 그 결과로 한 객체가 시뮬레이션 기간 동안에 전달해야하는 최종 통신 데이터 량은 OMM을 사용한 경우 감소한다.

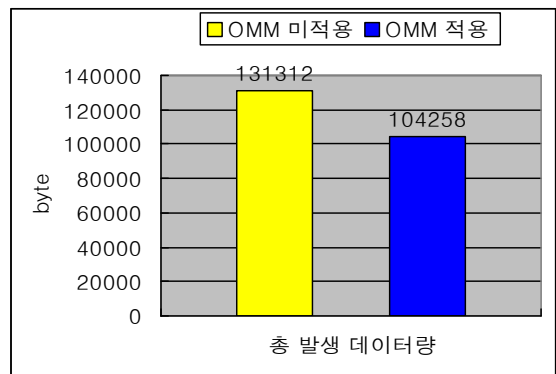


그림 10. 시뮬레이션 기간 동안 발생한 데이터량
Fig. 10. Quantities of communication data during Simulation

[그림 10]과 같이 시뮬레이션 수행동안 발생한 통신 데이터 량은 OMM을 적용한 경우에 더 적게 발생하였다.

이 시뮬레이션을 통하여 제안한 OMM이 38% 정도의 통신 부하의 감소 효과를 나타내는 것을 알 수 있다.

V. 결 론

항공교통관제 시뮬레이션의 경우 HLA를 적용하는 경우, 불필요한 객체 속성 변화 정보를 지속적으로 송수신해야 되는 문제가 있다. 비행 중인 비행기의 경우, 위치 변화 정보가 주된 데이터 정보이다.

이와 같이 속성 변화가 적은 경우, HLA 기반 시뮬레이션을 구현할 때, 제안한 OMM을 사용함으로써 통신 부하를 경감하여 시뮬레이션 성능을 향상시킬

수 있을 것이다. 또한 항공교통관제 시뮬레이션 외에 HLA 기반 각종 항행 관련 분산시뮬레이션에서 대량 객체를 사용하고 각 객체의 속성수가 많은 대규모 시뮬레이션에 적용할 경우 효율적일 것이다.

앞으로 DDM(Data Distribute Management) 서비스와 연계하여 HLA 기반 분산 시뮬레이션의 성능을 향상시키기 위한 연구를 수행할 것이다.

참 고 문 헌

- [1] U.S Department of Defense, "High Level Architecture Interface Specification", Version 1.3, April 1998.
- [2] 황재준, 이규영, 최재영, “분산 컴퓨팅 환경에서 HLA를 이용한 시뮬레이터 통합” *한국정보과학회 가을 학술발표 논문집*, 제 32 제 2호, 2005.
- [3] IEEE 1516 "Standard for Modeling and Simulation High Level Architecture Framework and Rules", IEEE 2000
- [4] D Frederick Kuhl, Richard Weatherly, Judith Dahmann, "Creating Computer Simulation System - An Introduction to the High Level Architecture", *Prentice Hall*, 1999
- [5] Judith S. Dahmann "High Level Architecture for Simulation", *Distributed Interactive Simulation and Real Time Applications*, First International Workshop, Jan. 1997.
- [6] IEEE 1516.2 "Standard for Modeling and Simulation High Level Architecture-Federate Interface Specification“, *IEEE 2000*
- [7] IEEE 1516.1 "Standard for Modeling and Simulation High Level Architecture - Object Model Template Specification", *IEEE 2000*
- [8] 김세환, 채수환 “HLA/RTI 기반 분산시뮬레이션의 객체 관리 성능향상” *한국정보처리학회 춘계 학술대회 논문집*. 2005.
- [9] Design, Implementation and Performance of the STOW RTI Progtoty. (RTI-s) James O.CAlvin, Carol J.Chiang, Stephen M.McGarry, Steven J.Rack,

Dantel J.Van Hook MIT Lincoln Lab.

김 명 규 (金明奎)



2002년 한국항공대학교 컴퓨터공학과 졸업(공학사)
 2004년 한국항공대학교 컴퓨터공학과 졸업(공학석사)
 현재 한국항공대학교 컴퓨터공학과 박사과정
 관심분야 : High Performance Computing, 자연어 처리 등임

김 정 호 (金廷好)



2008년 한국항공대학교 컴퓨터공학과 졸업(공학사)
 현재 한국항공대학교 컴퓨터공학과 석사과정
 관심분야 : 자연어 처리 등임

채 수 환 (蔡洙煥)



1973년 한국항공대학교 항공전자공학과 졸업(공학사)
 1985년 미국 Univ. of Alabama 전자계산학과 졸업(공학석사)
 1988년 미국 Univ. of Alabama 전기공학과 졸업(공학박사)
 1989~현재 한국항공대학교 항공전자정보통신공학부 교수
 관심분야 : 컴퓨터 구조, 병렬처리 시스템 등임