

HLA/RTI 구현을 위한 시스템 성능 평가

김 호경, 김 명선, 정 창성

e-mail: {hkim, kms}@snoopy.korea.ac.kr, csjeong@charlie.korea.ac.kr

고려대학교 전자공학과 분산/병렬처리 연구실

Tel. (02) 927 - 7683

Fax. (02) 926 - 7620

Performance Evaluation for Implementing HLA/RTI

HoKyung Kim, MyeongSun Kim, ChangSung Jeong

Distributed/Parallel Processing Lab.

Korea Univ Electronics Engineering

미국 국방성에서는 고수준 아키텍처(High Level Architecture)를 국방성의 모델링, 시뮬레이션 프로그램의 표준으로 채택하고 있으며, 광범위한 어플리케이션, 컴퓨팅 아키텍처, 통신 환경으로 그 영역이 확장되고 있다. 지난 '99 춘계 학술 대회에서 발표된 논문에서는 이러한 경향을 수용하면서 효율적인 분산 시뮬레이션을 지원하기 위한 시스템 아키텍처와 프로토콜을 제안하였다. 본 논문에서는 제안된 시스템의 변경 내용을 소개하고, 시스템의 프로토타입을 구현한 후, 그것에 대한 성능 평가 결과를 제시한다. 성능 평가는 시스템의 확장성과 네트워크 트래픽이 많은 경우의 메시지 전송 정책에 초점을 두고 진행되었다.

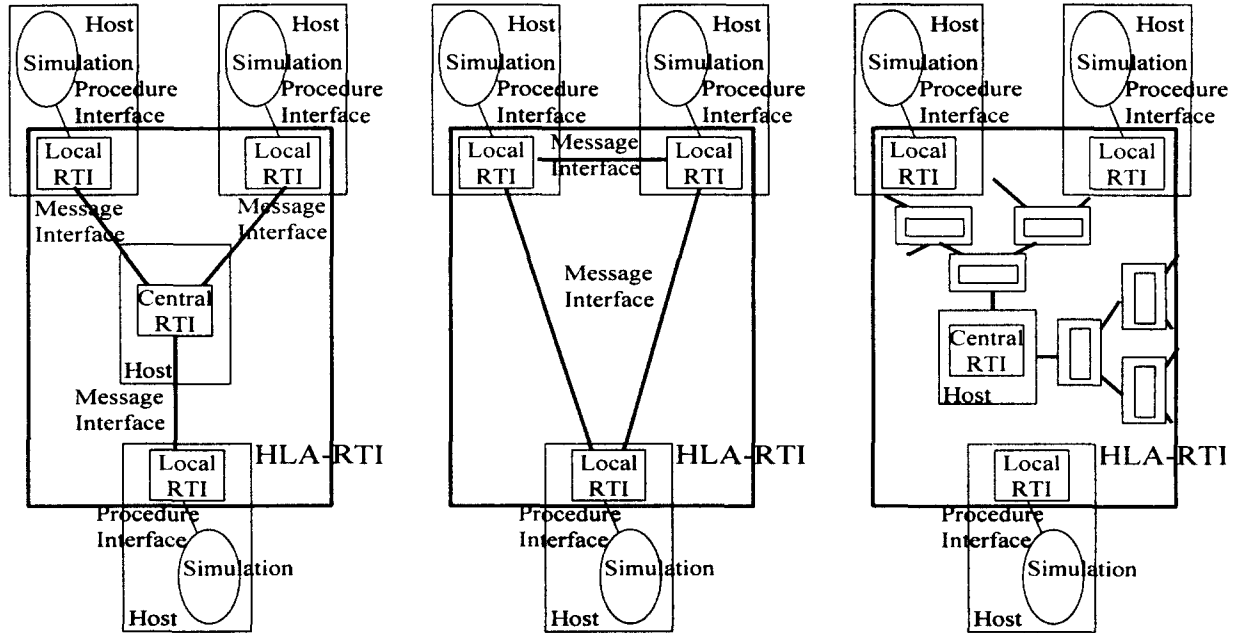
1. 서론

미국 국방성에서는 고수준 아키텍처(High Level Architecture)를 국방성의 모델링, 시뮬레이션 프로그램의 표준으로 채택하고 있다. 고수준 아키텍처는 시뮬레이션들 간에 상호 운용성(Interoperability)을 제공하고, 그 콤포넌트들의 재사용을 촉진시키는 데 그 목적이 있다. 이러한 표준화 활동은 실제로 미국 국방성의 산하 기관인 DMSO(Defense Modeling & Simulation Office)에서 주도하고 있기 때문에, 고수준 아키텍처를 언급할 때에는, 그것을 이용한 국방 관련 어플리케이션만을 고려하거나, 또는 고수준 아키텍처를 이용한 시뮬레이션들을 주로 생각하는 경향이 있다. 그러나 고수준 아키텍처는 분산/병렬

시뮬레이션, 통신 그리고 네트워킹 기술의 집약체이다.

이런 면에서 볼 때 고수준 아키텍처를 분산 시뮬레이션을 위한 네트워크 운영체제로 생각할 수도 있다.

위와 같은 특성으로, 고수준 아키텍처는 광범위한 어플리케이션, 컴퓨팅 아키텍처, 통신 환경으로 그 영역이 확장되고 있다. 지난 '99 춘계 학술 대회에 발표된 논문[10]에서는 앞에서 언급한 고수준 아키텍처의 특성을 수용하면서 효율적으로 분산 시뮬레이션을 지원하기 위한 시스템 아키텍처와 프로토콜을 제안하였다. 제안된 시스템



[그림 1] HLA-RTI 구현 모델

아키텍처에서 중점을 두었던 것은 시스템의 확장성(Scalability)과 관련된 문제였다. 이 문제를 해결하기 위해, 앞선 논문에서는 시스템을 서버간에 메시지를 전송할 때, 완전 이진 트리 형태의 메시지 흐름을 갖도록 설계해서 시뮬레이션 노드들의 수가 증가함에 따른 네트워크 지연을 최소화 할 수 있도록 설계하였다. 또한 전송된 메시지에 대한 ACK는 트리 경로상의 자식(child) 노드에서 받게 하였다.

다음 2장에서는 RTI 구현 모델을 나열하고, 그것이 [10]에 제시된 시스템에 어떻게 적용되었는지 설명한다. 3장에서는 프로토타입된 시스템과 내부 컴포넌트들의 기능, 관계를 이야기하고, 4장에서는 프로토타입된 시스템의 성능을 분석하고 평가한다. 마지막으로 제 5장에서는 결론과 향후 연구 방향에 대한 언급으로 글을 맺는다.

2. 관련된 연구

HLA의 RTI를 구현하기 위한 구조에 대한 모델은 크게 세가지로 나누어 볼 수 있다. [9] [그

림 1]은 이러한 세가지 타입의 구현 모델을 보여주고 있다.

[그림 1]의 (a)는 서버가 논리적으로 집중되어 있고, 물리적으로도 집중되어 있는 모델이다. [그림 1]의 (b)는 논리적으로도, 물리적으로도 분산되어 있는 모델이다. 다음 섹션에서는 이러한 중앙 집중적인 구조와 분산된 구조의 특징들에 대해 설명한다.

2.1 중앙 집중적인 서버 구현 모델

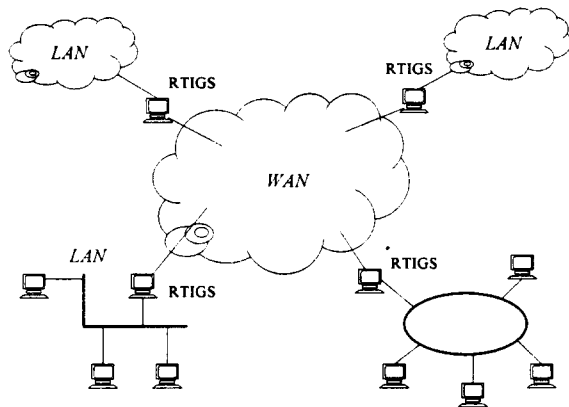
중앙 집중적인 서버 모델에서는 HLA가 전형적인 클라이언트/서버 모델로 정의되며, HLA 인터페이스는 페더레이트와 RTI간의 서비스를 정의하게 된다. 이 경우 페더레이트간의 직접적인 통신은 이루어지지 않게 된다. 이 모델은 구조적으로 가장 단순하기 때문에 구현하기가 용이하고, RTI의 행동이나 성능을 분석하기 쉬운 장점이 있다. 반면에 확장성(Scalability)이 떨어지는 단점이 있다.

2.2 분산된 서버 구현 모델

물리적으로 분산된 서버 구현 모델은 논리적인 분산 여부에 따라 [그림 1]의 (b), (c)와 같이 두

가지로 나뉘어질 수 있다. [그림 1]의 (b)는 논리적으로 분산된 구현 모델로서, 시스템에 시간 관리와 같은 전체적인 동작을 통제하는 호스트는 없다. 이러한 구조로 구현된 시스템으로는 RTI-s[8]가 있다. [그림 1]의 (c)는 시간 관리와 같은 전체적인 동작을 지원할 수 있는 중앙 서버가 존재하기 때문에 논리적으로 중앙 집중적인 서버 구현 모델이 된다. 이 모델은 이러한 중앙 집중적인 요소가 존재하지만 시스템의 확장성을 위해서 많은 하위 RTI들로 구성되어 있다. [그림 1]의 (c)는 중앙 RTI와 하위 RTI들이 이진 트리를 사용해서 구성된 예를 보여주고 있다.

2.3 제안된 시스템 아키텍처



[그림 2] 본 시스템의 네트워크 환경

[10]에서 제시된 시스템 아키텍처는 [그림 2]와 같은 네트워크 환경을 이루고 있다. [그림 2]에서 RTIGS와 LAN상의 호스트간의 관계는 [그림 1] (a)의 클라이언트/서버 구조를 이루게 된다. LAN 상에서 시물레이션을 수행하는 호스트들이, 자신이 속한 LAN에 있는 호스트들이나 또는 WAN을 통해 연결되어 있는 다른 LAN상의 호스트들과 통신하기 위해서는, RTIGS를 경유해야 한다. 또한, RTIGS들은 [그림 1]의 (b)와 (c)의 형태를 동시에 띄고 있다. 우선 RTIGS들은 물리적으로 분산되어 있으며, 중앙 RTI가 없기 때문에 논리적으로도 분산되어 있는데 이러한 특징은

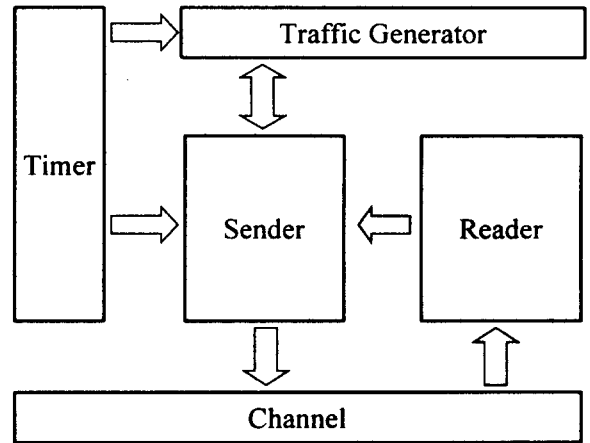
[그림 1]의 (b)의 구조와 유사하다. 그리고, 메시지 전송시 RTIGS들이 논리적으로 완전 이진 트리 구조를 형성한다는 점에서는 [그림 1]의 (c)의 구조와 유사성을 띄고 있다. 이와 같은 방식으로 본 시스템은 확장성(Scalability)을 확보할 수 있게 된다.

3. RTIGS의 내부구조

구현된 RTIGS의 내부 구성은 [그림 3]에 나와 있으며, 화살표의 방향은 메시지의 흐름을 나타낸다.

3.1 Traffic Generator의 기능

[그림 3]에서 Traffic Generator는 LAN(Local Area Network)을 시물레이션 하는 곳이다. 이것의



[그림 3] RTIGS의 내부구조

기능은 Timer에서 오는 신호에 따라 메시지들을 발생시킨 후, 그것을 Sender에게 보내는 것이다. 그 외에도 Sender로부터 어떤 메시지를 받을 경우, 그것을 소멸시키는 역할을 하게 된다.

3.2 Reader

Reader는 네트워크로부터 받은 메시지를 입력 큐(Queue)에 넣어서, 그것들을 Sender에게 전달하는 역할을 한다. 그런데 Reader에서는 네트워

크로부터 어떤 메시지를 받았는데, 입력 큐(Queue)에 그 메시지를 넣을 여유 공간이 없을 경우에는, 다음과 같은 메시지 전송 정책을 취하게 된다.

우선 받은 메시지가 신뢰 전송을 요구하는 메시지일 경우에는 입력 큐에 있는 최선 전송 메시지를 제거한 후, 네트워크로부터 받은 신뢰 전송 메시지를 입력 큐에 넣게 된다. 그런데 입력 큐에 최선 전송을 요구하는 메시지가 없을 경우에는 받은 메시지를 무시하게 된다. 만약 입력 큐에 여유 공간이 없을 때, 최선 전송을 요구하는 메시지를 받는 경우에는, 그 메시지는 바로 무시되게 된다.

이와 같이, Reader에서는 최선 전송을 요구하는 메시지들보다 신뢰전송(Reliable Delivery)을 요구하는 메시지들에게 우선 순위를 두는 메시지 전송 정책을 취하고 있다. 이러한 기능은 [10]에서 제시되었던 메시지 전송 정책에 추가된 부분으로서, 이와 같은 입력 큐 관리를 통해서 메시지 트래픽이 많아질 경우 신뢰 전송을 요구하는 메시지들을 보내는 데 더욱 우선 순위를 두게 된다.

3.3 Sender

Sender는 RTIGS의 핵심 모듈로서 Traffic Generator와 Reader로부터 오는 메시지들을 처리하는 곳이다. 이곳에서는 자신이 받은 메시지가 다른 RTIGS로부터 온 것이라면, 그것을 Traffic Generator에게 넘겨준다. 그리고 자신의 인덱스와 받은 메시지의 소스(Source) 인덱스로부터 자식 서버의 인덱스를 계산한 후, 전송 트리 내에서 그것이 존재할 경우, 받은 메시지를 자식 서버에게 전송해 준다. 자식 서버의 인덱스 계산은 완전 이진 트리의 구조적 속성을 이용하였다. 또한, Sender에는 신뢰 전송 (Reliable Delivery) 윈도우와 최선 전송(Best-Effort Delivery)윈도우가 구현되어 있어서, 메시지들에 대한 흐름 제어가

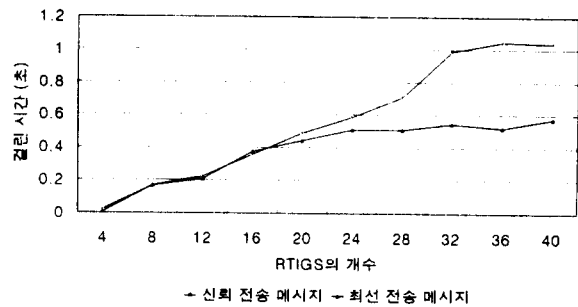
수행된다. 자식 서버의 인덱스 계산 방법과 전송 윈도우에 대한 내용은 [10]을 참고하기 바란다.

4. 실험 결과

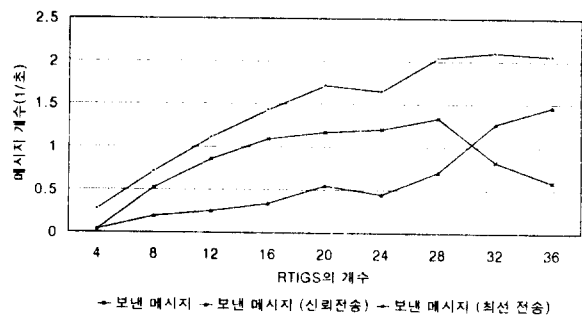
4.1 실험 환경 및 결과

실험은 두 대의 컴퓨터에서 이루어졌으며, 이 두 대의 컴퓨터는 LAN(10Mbps Ethernet)으로 연결되어 있다.

실험은 본 시스템의 확장성을 확인하는 데에 초점을 두었으며, 다음의 두가지 측면에서 이루어졌다. 우선 첫 번째 실험은 RTIGS의 수를 증가시키면서 전송 트리내에서 루트(root)로부터 트리의 말단(leaf)까지 메시지가 도달하는 데 걸리는 시간을 측정하였다. 두 번째 실험은 RTIGS의 수를 증가시키면서 각각의 RTIGS가 단위 시간당 전송하는 메시지의 개



[그림 4] 메시지 도착 지연 시간



[그림 5] 단위 시간당 보낸 메시지

수가 어떻게 변화하는가를 알아보았다.

4.1.1 메시지 도착 지연 시간 측정

[그림 4]에서 가로축은 RTIGS의 개수를 나타내고, 세로축은 메시지 도착 지연 시간을 나타낸다. 실험에 사용한 두 대의 컴퓨터에 있는 타이머의 동기(synchronization)를 맞추기가 어렵기 때문에 한 대의 컴퓨터만을 이용해서 시간을 측정하였다.

[그림 4]에서도 알 수 있듯이 시간 증가 속도는 신뢰 전송 메시지가 최선 전송 메시지보다 더 빠르다는 것을 알 수 있다. 즉, 신뢰 전송 메시지의 도착 지연 시간이 최선 전송 메시지의 도착 지연 시간보다 크다는 것을 알 수 있다. 이와 같이 신뢰 전송 메시지의 지연 시간이 더 큰 이유는, 각각의 RTIGS가 신뢰 전송 메시지를 받으면, 상위 서버에게 확인(Acknowledgement)을 보낸후 메시지를 하위 서버에게 전송하기 때문이다.

[그림 4]에서 메시지 도착 지연 시간의 증가는 RTIGS의 개수에 비례해서 증가하고 있다. 즉, RTIGS의 개수에 대해 메시지 도착 시간이 선형적으로 변화하고 있다. 이런 면에서, 제안된 시스템 아키텍처는 시스템의 규모가 커지더라도, 메시지의 도착 지연 시간이 그렇게 크게 증가하지 특성을 확인할 수 있었으며, 이러한 면은 시스템이 확장성을 갖는 데 큰 기여를 할 수 있다.

4.1.2 단위 시간당 전송한 메시지의 변화

[그림 5]은 RTIGS의 개수를 증가시킬 때, 각각의 RTIGS가 단위시간당 보내는 메시지 개수를 그래프로 나타낸 것이다. 최선 전송 메시지와 신뢰 전송 메시지의 생성 비율은 4 : 1 이며, 네트워크에서의 손실율은 영(zero)으로 설정했다.

[그림 5]에서 단위 시간당 보낸 메시지의 증가 경향도 선형성을 보여주고 있으며, 한가지 주목할 점은, RTIGS가 30개인 경우를 전후로 해서 두 곡선이 서로 교차하고 있다. 이와 같은 현상

이 나타나는 이유는, 메시지 트래픽이 많아서 Reader의 입력 큐(Queue)에 여유 공간이 없을 경우, Reader에서는 Sender에게 메시지를 보낼 때 최선 전송 메시지보다는 신뢰 전송 메시지를 보내는 데에 더 우선 순위를 두기 때문이다.

5. 결론 및 향후 연구

본 연구에서 제안된 시스템은 중앙 실시간 하부구조를 갖지 않는 분산된 서버 구현 모델을 바탕으로 하고 있으며, 제안된 시스템 아키텍처는 실험 결과에서도 알 수 있듯이, 규모면에서 상당한 확장성을 가지고 있음을 확인할 수 있었다.

본 연구에서는 이러한 결과를 바탕으로 고수준 아키텍처의 실시간 하부구조에 대한 설계가 진행될 계획이다.

[참고 문헌]

- [1] V. Jacobson, "Congestion Avoidance and Control," *Proceedings of ACM SIGCOMM'88*, Pages 314-328, August 1988.
- [2] R.Yavatkar, J. Griffioen, M. Sudan. "A Reliable Dissemination Protocol for Interactive Collaborative Applications", *Proceedings of ACM Multimedia*, Pages 333-344, 1995
- [3] J.C. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol," *Proceedings of IEEE INFOCOM '96*, Pages 1414-1424, March 1996
- [4] B.N. Levine, and J.J. Garcia-Luna-Aceves, "A Comparison of Known Classes of Reliable Multicast Transport Protocols," to appear in *Proceedings of International Conference on Network Protocols*, October 1996
- [5] U.S. Defense Modeling and Simulation Office, "High Level Architecture Object Model Template Specification", Version 1.3, 5 February 1998
- [6] U.S. Defense Modeling and Simulation Office, "Department of Defense High Level Architecture

Rules", Version 1.3, 5 February 1998

[7] U.S. Defense Modeling and Simulation Office,
"High Level Architecture Interface Specification
Draft", Version 1.3, 25 April, 1998

[8] James O. Calvin, Carol J. Chiang, Stephen M.
McGarry, Steven J. Rak, Daniel J. Van Hook
"Design, Implementation, and Performance of the
STOW RTI Prototype (RTI-s) *In Proc of the
Spring Simulation Interoperability Workshop, 1997*

[9] Masakazu Furuichi, Masaji Mizuno, Hiroyuki
Miyata, Minoru Miyazawa, Satoshi Matsumoto,
Kazuhiro Aoyama " Performance Evaluation Model
of HLA-RTI And Evaluation Result of eRTI" *In
Proc of the Fall Simulation Interoperability
Workshop, September 1997*

[10] 김호경, 김명선, 정창성 "HLA/RTI 구현을 위한
분산 시스템 구조" 한국 시뮬레이션 학회 '99 춘계 학
술대회, 1999