ภาคผนวก

ภาคผนวก ก บทความเรื่อง An Approach of Network Simulation Tool for Analyzing the HLA Performance

The 3rd International Symposium on Communications and Information Technology, 2003.

An Approach of Network Simulation Tool for Analyzing the HLA Performance

Atinat Palawan, Pichaya Tandayya, and Suntorn Witosurapot

Computer Engineering Department, Faculty of Engineering,
Prince of Songkla University, Hadyai, Songkhla, 90112 Thailand
Email: s4412081@maliwan.psu.ac.th, pichaya@coe.psu.ac.th, wsuntorn@coe.psu.ac.th

Abstract

This paper presents an approach for analyzing the High Level Architecture (HLA) performance tool base on the NS simulation tool. This approach will allow users to configure network models of distributed simulation federation. So that, federation developers can observe the performance of distributed simulation federation in such a way that in a dynamic environments e.g., the difference number of objects on each nodes and network topology can be monitored. In the previous works proposed by the HLA research community, rather focused on external parameters of the actual distributed simulations by looking at the overview of the systems. This tool will benefit the federation developers in testing and diagnosing the performance problems and in optimizing the performance of the HLA federations.

Keywords:
Network Simulation Tool, Federation Performance, Distributed Simulation, HLA, RTI

1. Introduction

At the early development of distributed simulation using High Level Architecture (HLA), the developers often experienced performance problems such as overload in processing nodes or high consumption of the bandwidth of communication links that were very hard to discover the main cause [1]. Generally, HLA federations are usually very complex so that the symptoms of the performance problem usually appear at a point that far from the source of the problem.

Over the past few years, many tools, techniques and models about HLA have been developed for studying, understanding and solving the performance problems of HLA federations [2]. They mainly focused on external and simple parameters of the actual distributed simulations such as constant values and constant formula that gave only the overview performance of the systems.

None of them concentrated on a federation simulation tool that enables users to easily configure network models of distributed simulations and observe the performance of distributed simulation systems as the topology of the systems and the number of objects on each node changes.

This paper proposes an approach to develop a simulation tool for HLA distributed simulation developers that enables them to test and diagnose the performance problems and to optimize the performance of the HLA federations.

In the following section, section 2 will describes about some background of HLA and its performance models, section 3 about the simulation tool we chose-NS, section 4 involve with how do we add the models to NS and the discussion in the final section.

2. HLA Background

The High Level Architecture (HLA) has been developed by the U.S. Department of Defense (DoD) for reusability and interoperability in distributed simulation area [3, 4]. The HLA is described as a rule and specification. In addition, DoD and their partners have implemented the Runtime Infrastructure (RTI) as a middleware for cooperate between federates as shown in Figure 1.

A HLA federation composes of three sub-components as shown in Figure 2.

1. network model
2. simulation model
3. RTI model

The network model includes a representation of a network topology, link delay, queue types, protocols and so on. The simulation model includes a group of entity models that represent the statistical characteristic of each entity. We can consider a simulation entity as a random message generator that produces data or events [2].

The RTI model is based on the HLA Interface Specification version 1.3 NG [5]. The RTI model includes a representation of the RTI service functions such as create, join, publish, subscribe, update, reflect, etc. All RTI services provide the interface protocols between the simulation model and the RTI model.

In this paper, we will focus on the design of the simulation tool composed of the three sub-models about how the HLA federation developers can use the simulation tool for observing and predicting the performance and for solving the performance problems of HLA federations.

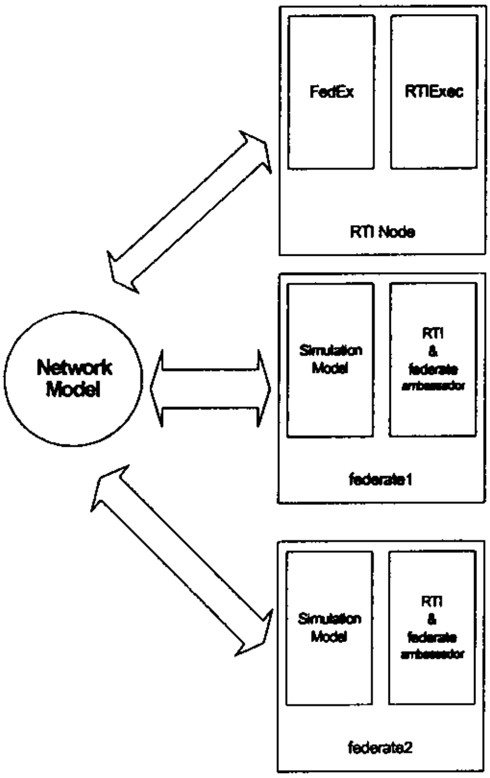

Figure 1. Concept of an HLA federation

3. Network Modeling with NS

The Network Simulator (NS) has been chosen as our base simulation platform since it is well known in the network research community, well documents and was specifically build for network analysis [6]. NS is a discrete event simulator targeted at networking research and provides substantial support for simulations of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. It was designed with an object-oriented style that the user can extend or add in functions by deriving any existing components in order to create as a new protocol or a new network entity.

Generally, other works had proposed only the HLA simulation and RTI models and did not focus on the simulation of the network model. The network model was represented by only constant parameters or formula [1, 5]. NS is an efficient network simulation tool that can be used to represent a complex network topology including essential parameters such as link delay, queue types and so on. The HLA simulation and RTI models will also be added into NS in the same way that existing network modules have been represented. This approach will lead us to a network model that can simulate an HLA federation with a complex and changeable network topology and network parameters as shown in Figure 3. The users can then add a few lines of code in order to change the configuration of simulation scenarios and observe the performance on case studies of HLA federations.

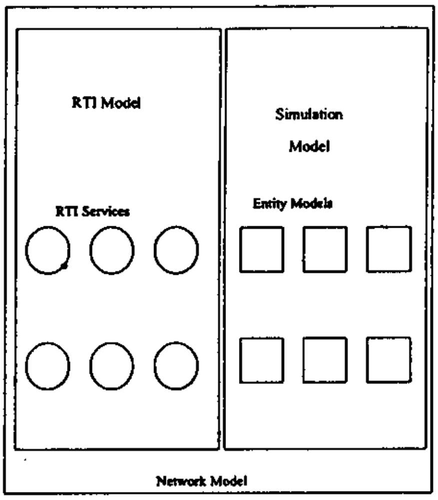

Figure 2. Three sub-components of HLA federations

4. Design

As we mentioned in the previous section, we use NS as an infrastructure to represent the network model of HLA federations. In this section we will describes about the design or how we add the RTI model and simulation model into NS.

4.1 RTI model

The RTI employs all of the 6 RTI specification's management areas as follows.

- Federation Management
- Declaration Management
- Data Distribution Management
- Object Management
- Time Management
- Ownership Management

Each area composes of HLA services. Several previous works described how to collect the statistical characteristic data of the services and then conclude into a general form of models. Our performance model of RTI focuses on the update and reflect services of the Data Distribution Management (DDM) area as these services are explicitly demonstrated [5] and they significantly affect the performance of HLA federations [7].

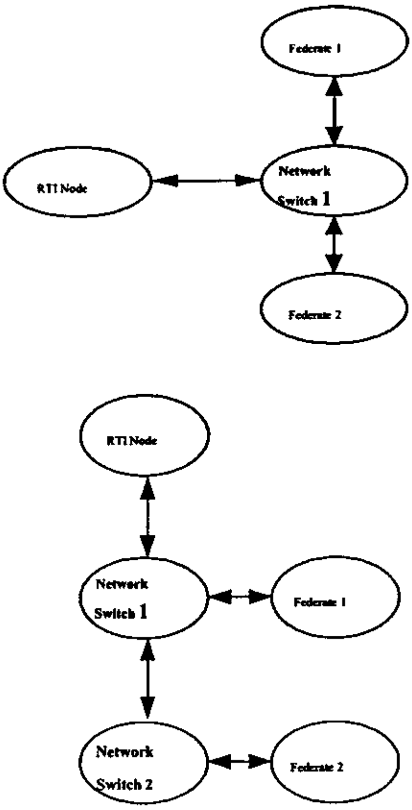

Figure 3. Different network topologies cause different performance results

The DDM area specified for the HLA is a data reduction mechanism that decreases total data sent and received for large scale distributed simulations [7]. The publish and subscribe mechanisms are used in the RTI to filter messages before forwarding them to subscribed federates. However, they introduce more bandwidth consumption on the RTI node. The RTI model are represented in term of time consuming update and reflect services at the RTI node as listed below [2].

Total time in RTI node:

$$T = T_{UPDATE} + T_{I/O} + MAX_{FED}(T_{REFLECT}) \quad (1)$$

T_{UPDATE} is the processing time when update event occurs at RTI node. The effect of update event is determined in term of query database delay such as federation object management database, federation execution database, subscribe database, object instance database. $T_{REFLECT}$ is the processing time when reflect event occurs at RTI node. Due to the time of transferring the message to time stamp order queue. $T_{I/O}$ is the overhead time of input/output devices. MAX_{FED} is maximum number of federate that RTI node reflects the messages to.

The federation developers then can use the existing HLA tools to capture data from the actual system and characterizes its expected value, shape and type of probability distribution before using the RTI model.

When the message update sent from a federate to the RTI node, it will produce the latency at the RTI node depending on the values of parameters of the above formula. These formula will be implemented into the NS as an entity that user can configure any variables with expected value and probability density function.

4.2 Simulation Model

The simulation model represents the behavior of simulation entities in term of average value of sent/received messages. The simulation model varies due to the characteristic of the simulation federation.

Due to the complexity of a federation, we adopt a probabilistic or statistical approach in characterizing the latency [2]. Generally, we do not explicitly address the chain of events or messages that was produced by any simulation entity at any federate because it will be too complex. Instead, we estimate the statistical values of sending and receiving messages among simulation entities such as expected values or means and the probability function.

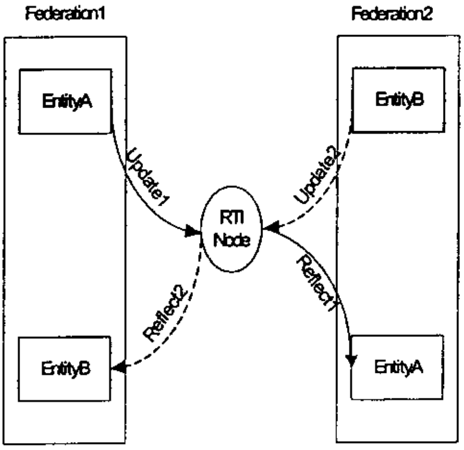

Figure 4. Flow of update/reflect services among entities.

For example, in Figure 4 if Entity A at Federate 1 sends or updates data to Entity A at Federate 2, this action will affect the followings:

- Processing delay of update1 on Federate 1.
- Link delay and bandwidth consumed between Federate 1 and the RTI node.
- Processing delay of update1/reflect1 on RTI node due to the RTI model.

- Link delay and bandwidth consumed between the RTI node and Federate 2.
- Processing delay of reflect1 on Federate 2.

The user uses can configure the relationship among simulations entities, processing load or delay of sending/receiving messages at federates and the latency due to update/reflect services at the RTI node.

4.3 Additional HLA for NS

NS provides several network modules in an objects oriented style that users can add new protocol into it. Existing application protocols are, for example such as FTP, TELNET and so on. Our HLA module will derive an application level protocol that will interact with the transport protocol. From the class hierarchy shown in Figure 5, we add the HLA module as a derived class of the Application class in order to expand the ability of the HLA specific behaviors.

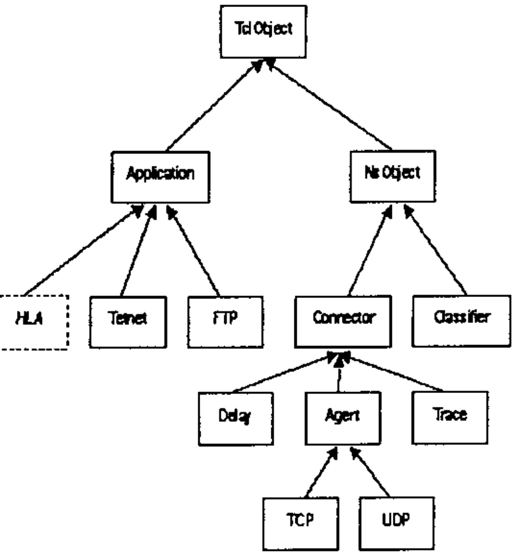

Figure 5. Class hierarchy of modules in NS

As shown in Figure 6, if we consider the relationship among application level and transport level, the HLA module can be attached and interact with UDP, TCP or others and it will acts the behavior due to the underlying protocol used. If the users configure the simulation to used UDP protocol so they can use multicast routing ability already implemented in NS.

5. Performance Measurement and Evaluation

In order to prove our idea and demonstrate the different network performance on different network topologies, we simulated the two HLA federations as shown in Figure 3 in NS. Topology 1 referred to a federation with only one network switch. Topology 2 referred to a federation with two network switches. We developed an RTI module and assigned a node to play the role of RTI. We also developed object sources that produced UDP flows and assigned two nodes to act as Federate 1 and Federate 2. The object in Federate 2 behaved like a shadow of the object in Federate 1. Therefore, there was only one flow of data distribution, from Federate 1 to Federate 2.

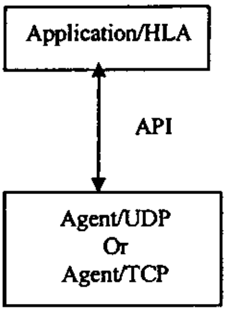

Figure 6. The relationship among Application level class and underlying class.

We determined exponential distribution means as follows: 1 packet/second for the arrival rate of distributed data and 110 bytes for the size of packets. Each transmission link between each node had the propagation delay of 10 ms and the bandwidth of 1 k bps. With this configuration, we observed the performance of the two topologies and ignored the delay at the RTI node.

The two factors we observed were *queue delay* and *average queue length* represented at each node as shown in Table 1 and Table 2. The total delay was the summation of queue delays, transmission link delays, propagation delays between Federate 1 and Federate 2. The total average delay of Topology 1 was shorter than the total delay of Topology 2 (19.451 sec to 23.928 sec).

Table 1. The delays in the network queue.

Queue Delay	Topology 1 (sec)	Topology 2 (sec)
Federate 1 to Switch 1	6.283	6.283
Switch 1 to RTI node	2.910	2.910
RTI node to Switch 1	3.236	3.236
Switch 1 to Federate 2 or Switch 1 to Switch 2	3.462	3.462
Switch 2 to Federation 2	-	3.637
Total Queue Delay	15.891	19.528

The 3^{rd} International Symposium on Communications and Information Technology, 2003.

Table 2. The average queue length at nodes.

Average Queue Length	Topology 1	Topology 2
Federate 1 to Switch 1	6.27	6.27
Switch 1 to RTI node	2.90	2.90
RTI node to Switch 1	3.22	3.22
Switch 1 to Federate 2 Or Switch 1 to Switch 2	3.45	3.45
Switch 2 to Federation 2	-	3.62

The unequal delay of different topologies obviously affects the speed of the HLA federation. In HLA, data distribution plays an important role to the speed of the federation. From the result shown in Table 1, we could estimate the proportional speed of data distribution between the two topologies that Topology 1 was faster than Topology 2 18.710 percent. If there are more network switches between Federate 1 and Federate 2, the accumulated total average delay will be much distinct.

The queue length at each node suggests about the blocking probability at each node. Longer average queue length has higher probability for dropping packets. This fact leads to the conclusion that Topology 1 has more reliability than Topology 2 when the traffic is congested as the total queue length of Topology 1 is shorter than that of Topology 2 (Table 2).

The result suggest that the two factors, queue delay and average queue length, are important factors for analyzing the network performance of an HLA federation. In our future work, more complex case studies and more factors are to be analyzed and presented.

6. Conclusion

This paper has presented the initial stage of the development of a federation simulation tool that will help distributed simulation developers studying, understanding and solving the performance problems of building HLA federations.

We used a network simulation tool, NS, and presented some testing results on a case study. The results showed that our approach is in the right direction and that further development will be able to analyze performance and suggest the solutions to the design of HLA federations.

6. References

[1]. Duncan C. Miller, Steven B. Boswell, "A General Framework for Modeling Federation Performance", 2001. 01S-SIW-070.

[2]. Stephen R. Kolek, Steven B. Boswell, Harry M. Wolfson, "Toward Predictive Models of Federation Performance:Essential Instrumentation", Fall Simulation Interoperability Workshop,2000.00F-SIW-085.

[3]. Defense Modeling and Simulation Office Web Site, http://hla.dmso.mil.

[4]. Richard Fujimoto, Peter Hoare, HLA RTI Performance in High Speed LAN Environments. Fall Simulation Interoperability Workshop, 1998. 98F-SIW-085.

[5]. Jenifer McCormack, Cristl Weckenman, Gene Lowe, "Development of a HLA Constructive Performance Model", Spring Simulation Interoperability Workshop, 1999.99S-SIW-146.

[6]. ns manual, The VINT Project, http://www.isi.edu/nsnam/ns/doc/

[7]. Katherine L. Morse, Michael Zyda, "Multicast grouping for data distribution managemen", Simulation Practice and Theory 9, 2002. p121-141