LoI-Based Flow Control on Low-Bandwidth Federates

Zhong Zhou and Qinping Zhao

State Key Laboratory of Virtual Reality Technologies
School of Computer Science and Engineering, Beihang University,
Beijing 100083, P.R.China
zz@vrlab.buaa.edu.cn

Abstract. With the development of simulation scalability, federates of various abilities are required to cooperate in a virtual environment. Restricted to HLA(High Level Architecture) standard, all the subscribing federates have to process the same traffic from an virtual object. As a result, existing RTI(Run-Time Infrastructure) software cannot conduct flow control on low-bandwidth federates. This paper suggests to reduce the object updating frequency and to restrict the update interval based on LoI(Layer of Interest) according to the host bandwidth. A bandwidth adaptive flow control model was built on the basis of LoI. This model depicts the way to conduct flow control with fidelity guarantee. Flow control algorithms are proposed to drop update packets according to the object LoI and packet LoI. Experiments illustrated that the flow control model can cut down the federate bandwidth requirements.

1 Introduction

Large-scale distributed simulation consists of thousands of objects moving and interacting nearly all the time. Each object usually need to send tens of attribute value update packets per second while moving or interacting. Thus quantities of packets are produced in the simulation. It results in the fact that some low-bandwidth hosts cannot treat the traffic in distributed simulations of even only tens of objects. It's very important to support low-bandwidth hosts in distributed simulation because most portable devices are wireless with low-bandwidth.

High Level Architecture (HLA) is the newest standard of distributed simulation. It was initiated by US DoD Defense Modeling&Simulation Office (DMSO) to facilitate the interoperability and reuse of simulators [1]. HLA standard consists of three parts, framework and rules[2], object model template (OMT)[4] and interface specification[3]. It prescribes a software named Run-Time Infrastructure (RTI) to provide services with APIs conformed to the interface specification. RTI decides to a large extent the simulation scalability and efficiency. Because RTI is responsible for the data delivery of federates, the key point of supporting low-bandwidth federates lies in how to conduct flow control in RTI.

We had brought forward a theory of LoI(Layer of Interest) in previous work[13-15]. LoI defines a relevance classifier based on the impact of spatial distance on receiving attributes and attribute values. An extension to HLA is also provided by LoI. This paper proposed a flow control model on the basis of LoI. Assume a federate updates attribute value of an object in its activity. With the help of LoI, some low-bandwidth federates can select a small portion of updates in a lower frequency to process according to the bandwidth without inconsistency.

The rest of this paper is organized as follows. Section 2 introduces the related work. The LoI related concepts are stated in Section 3. Section 4 describes the flow control model based on LoI and flow control algorithms. Experiment evaluation is presented in Section 5. Finally, the paper concludes with Section 6.

2 Related Work

With the development of distributed simulation, HLA has been criticized in many aspects such as limitation in transportation type[5]. It regulates only two transportation types, reliable and best-effort. This rough classification is insufficient for federates with various abilities. Most RTIs are too strictly restricted to HLA standard to overcome this inherent limitation of HLA, such as DMSO RTI NG[6,7], pRTI[8] and Mak RTI[9]. By far, few RTIs can conduct federate flow control and support low-bandwidth federates in distributed simulation with tens of objects. STOW RTI-s [10], which was developed for STOW(Synthetic Theater of War) exercises, is an exception. Its implementation differs much with HLA standard. A minimum rate service was adopted in RTI-s to meet the bandwidth and receiving ability of hosts. In the service, attributes are sent at a minimum rate no matter they have changed values or not.

There have been several studies in adjusting frequencies to lower the bandwidth requirements of distributed simulation. Zhou etc proposed a utility model[11] to evaluate the importance of a simulation entity. Based on the utility model, they devised some flexible update mechanisms of controlling the sending frequency to use the bandwidth more efficiently. We have implemented a distributed RTI – BH RTI [12, 13]. BH RTI was proved to be efficient in RTI congestion control using frequency adjustment based on LoI [14].

Minimum rate service is applying fixed minimum bandwidth usage on federates. Existing studies on lowering bandwidth requirements didn't take the federate physical bandwidth into account. The federate flow control model proposed in this paper conducts flow control according to different federate bandwidth requirements.

3 The Layer of Interest (LoI)

Existing publish-subscribe mechanisms in HLA standard can only judge whether a message is relevant to a subscriber or not. Aiming to solve the relevance evaluation problem, a relevance evaluation mechanism Layer of Interest (LoI) was proposed in our previous work [14][15]. LoI defines a relevance classifier based on the impact of spatial distance on receiving attributes and attribute values. It is divided into six layers on the receiver interest in objects. LoI is defined as an enumerate data type.

```
enum LoI { NO_LAYER = 0, LAYER_CRITICAL, LAYER_VISION,
LAYER_ ABOUT, LAYER_COMPONENT, LAYER_INSIDE }.
```

Some important LoI variables are listed in Table 1.

Table 1. Symbols in the LoI

Symbol	Definition
$P_m^{(i)}$	LoI of publisher over object class ***i*** with ***m***-size attribute set
$p_m^{(i,o)}$	LoI of local object instance ***o*** of object class ***i*** with ***m***-size attribute set
$\eta_j^{(i,o)}$	LoI of attribute update/reflect with ***j***-size attribute set of object instance ***o*** of object class ***i***
$S_k^{(i)}$	LoI of subscriber over object class ***i*** with ***k***-size attribute set
$s_l^{(i,o)}$	LoI of remote object instance ***o*** of object class ***i*** with ***l***-size attribute set

We drew two important deductions about the LoI relationship among the three LoIs, that of the publisher, of the subscriber and of messages.

Deduction 1. A publisher can only send attribute updates of LoI $\eta_j^{(i,o)} \leq p_m^{(i,o)}$.

Deduction 2. A subscriber can only receive attribute reflects of LoI $\eta_j^{(i,o)} \leq s_l^{(i,o)}$.

In the LoI-based publish-subscribe environment, the publisher works with $P_m^{(i)}$ of object class ***i*** and $p_m^{(i,o)}$ of local object instance ***o***. And a subscriber works with $S_k^{(i)}$ of object class ***i*** and $s_l^{(i,o)}$ of remote object instance ***o***. The four LoIs denote the dynamic detail relevance in publish-subscribe sides. Messages will be tagged with LoI $\eta_j^{(i,o)}$. $\eta_j^{(i,o)}$ represents the fundamental relevance of messages.

We use the symbols below in this paper to represent corresponding LoI in short.

```
L0 = LAYER_CRITICAL;
L1 = LAYER_VISION;
L2 = LAYER_ABOUT;
L3 = LAYER_COMPONENT;
L4 = LAYER_INSIDE.
```

4 LoI-Based Flow Control

The LoI-based flow control is proposed in the chapter including the architecture, system model and algorithms.

4.1 Federate Flow Control Architecture

RTI implementation has several architecture types. It's of benefit not to allow the federates communicating individually. To conduct flow control on federates, RTI had better take over communications with federates. The federate flow control architecture is shown in Figure 1. Several federates communicate with BH RTI. BH RTI establishes a flow control function for the federates, controlling federates' incoming and outgoing messages.

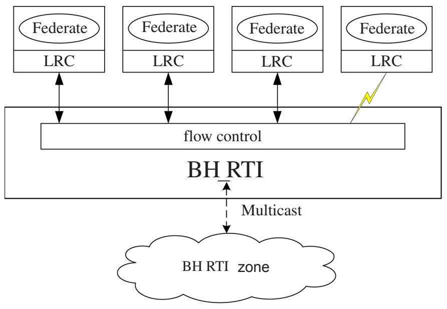


Fig. 1. Federate Flow Control Architecture

4.2 System Model

In a distributed virtual environment containing tens or even hundreds of objects, low-bandwidth federates are unable to receive the messages from so large remote objects. Because there's no relevance evaluation in previous work, all the messages are of the same essence to the subscribing federates. LoI evaluates the message relevance, and then provides a way for RTI to control the federates' data flow.

To reduce the bandwidth usage in low-bandwidth federates, it's necessary to cut down irrelevant messages as possible. A practical idea is to control the update frequency of each object based on its LoI according to the federate bandwidth. Now that each object has been attached with LoI, federates can reduce their sending and receiving messages. This reducing is based on the federate bandwidth. Low-bandwidth federates will send out less frequent messages than other federates. So is the message receiving. Here the update/reflect frequency is the frequency in which the federate actually sends out the object messages or receives the object messages. Because the reflect frequency and important messages are guaranteed, the system can run correctly although more messages are dropped.

The system model is illustrated as Figure 2. When an attribute update message of some object instance is received in RTI, RTI compares the message attributes with the object attribute subscription of the federate. According to the federate bandwidth, RTI

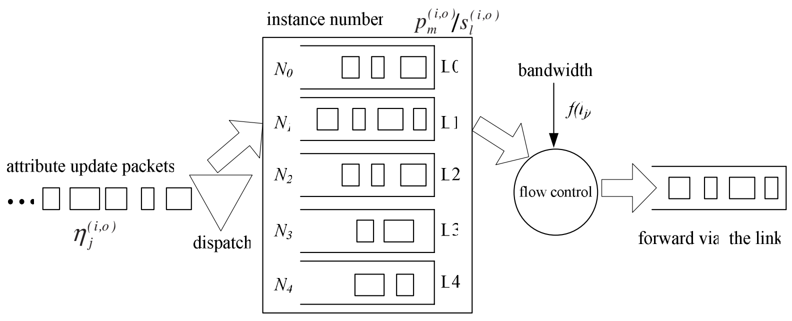


Fig. 2. Flow Control Model based on LoI

regulates the update/reflect frequency of objects in different LoI. The Attribute values of different LoIs are verified individually to guarantee the fidelity of objects.

Users should provide standard update frequencies of LoI according to the federate bandwidth. It's acceptable for an object instance to be reflected in the standard update frequency. There may be different setting in different federate because of their bandwidth or requirements.

Standard Update Frequency of LoI. *Standard update frequency of LoI is marked as f(l), where l is LoI of the object instance in the federate.*

Then we can get $f(l) = f(p_m^{(i,o)})$ for local object instance and $f(l) = f(s_l^{(i,o)})$ for remote object instance. Suppose there are n object instances of m types of object classes C_1, $C_2 \ldots C_m$ in the simulation. And there are n_i object instances of class C_i, where $i \in [1,m]$ and $\sum_{i=1}^{m} n_i = n$. In a simulation time, there are $\mathcal{N}_0$ instances of L0, $\mathcal{N}_1$ instances of L1, $\mathcal{N}_2$ of L2, $\mathcal{N}_3$ of L3 and $\mathcal{N}_4$ of L4, where $\sum^{4} \mathcal{N}_i = n$.

Let $U(I_j)$ be the required update frequency of object instance I_j for the federate to process correctly. In the regular mode, the total data traffic produced in the federate will be $\sum_{j=1}^{n} U(I_j) = n*U(I_j) = n*U_0$, where U_0 is the update frequency required in the simulation. It's obvious that $U_0 >= f(l)$.

The total outgoing data traffic in the federate will be

$$\sum_{j=1}^{n} f(l_j) = \sum_{j=1}^{n} f(p_m^{(i,o)})$$

$$= \mathcal{N}_0 U_0 + \sum_{j=1}^{N_1} f(L1) + \sum_{j=1}^{N_2} f(L2) + \sum_{j=1}^{N_3} f(L3) + \sum_{j=1}^{N_4} f(L4)$$

$$= \mathcal{N}_0 U_0 + \mathcal{N}_1\ f(\mathrm{L1}) + \mathcal{N}_2\ f(\mathrm{L2}) + \mathcal{N}_3\ f(\mathrm{L3}) + \mathcal{N}_4\ f(\mathrm{L4})$$

Suppose there are n' remote object instances in the federate and corresponding LoI instance counts are $\mathcal{N}_0'$, $\mathcal{N}_1'$, $\mathcal{N}_2'$, $\mathcal{N}_3'$ and $\mathcal{N}_4'$, where $\sum_{i=0}^{4} \mathcal{N}_i' = n'$, the total data traffic received in the federate will be $\sum_{j=1}^{n'} U(I_j) = n' * U(I_j) = n' * U_0$.

The total incoming data traffic in the federate will be $\sum_{j=1}^{n'} f(l_j) = \sum_{j=1}^{n'} f(s_l^{(i,o)})$

$$= \mathcal{N}_0' \mathrm{U}_0 + \sum_{j=1}^{N_1'} f(L1) + \sum_{j=1}^{N_2'} f(L2) + \sum_{j=1}^{N_3'} f(L3) + \sum_{j=1}^{N_4'} f(L4)$$

$$= \mathcal{N}_0' U_0 + \mathcal{N}_1'\ f(\mathrm{L1}) + \mathcal{N}_2' f(\mathrm{L2}) + \mathcal{N}_3' f(\mathrm{L3}) + \mathcal{N}_4' f(\mathrm{L4})$$

It can be seen from above that the bandwidth used by the object instances is divided into four groups plus bandwidth for critical messages. Those critical update/reflect messages won't be dropped. Object instances of higher LoI can be put more bandwidth on. In this way, we can conduct flow control on low-bandwidth federates by regulating update/reflect frequencies according to the federate settings.

4.3 Algorithms

Although some LoI terms have been used in our previous work, they are unsuitable for the flow control requirements introduced above. Another series of LoI terms should be defined to check the flow fidelity of all LoIs, so that federates won't miss the latest attribute update in flow control. We prescribe that attributes are associated with correct LoI values in HLA OMT extension. Then a federate will update attribute values in the frequency of its object LoI. Other federates, that are interested in the object instance, can select a small portion of updates in a relatively lower frequency to process without inconsistency.

We define the new series of LoI terms Flow LoI to depict the flow fidelity of all LoI values. The data type of Flow LoI is a BYTE, shown in Figure 3. The lower 5 bits are given to five LoI values in order, and higher 3 bits are reserved. The bit value, 0 or 1, represents whether the corresponding LoI value is used. When it's of the federate, it means whether the federate publish/subscribe attributes of the LoI value. When it comes to messages, it means the message contains attribute values in the LoI value.

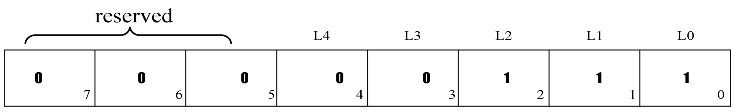


Fig. 3. The Data Type of Flow LoI

The federate preserves Flow LoIs for publishing local object instances and subscribing remote object instances. Flow LoI is evaluated according to the LoI value $p_m^{(i,o)}$ and $s_l^{(i,o)}$ as Figure 4. When the federate publishes/subscribes an object instance in a LoI value, messages of attributes in lower LoI values is certain to be relevant.

Symbol $p_f^{(i,o)}$ denotes the Flow LoI of federate's publishing object instance ***o*** of object class ***i***.

Symbol $s_f^{(i,o)}$ denotes the Flow LoI of federate's subscribing object instance ***o*** of object class ***i***.

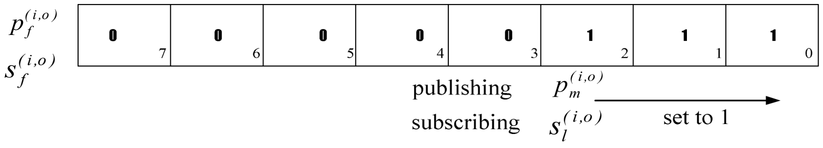


Fig. 4. Flow LoI of Object Publish/Subscription

The Flow LoI $p_f^{(i,o)}$ and $s_f^{(i,o)}$ is obtained from corresponding LoI $p_m^{(i,o)}$ or $s_l^{(i,o)}$. The evaluation of $p_f^{(i,o)}$ and $s_f^{(i,o)}$ is shown as Table 2.

Table 2. Flow LoI of Publish/Subscription: Evaluation

LoI $p_m^{(i,o)}$ or $s_l^{(i,o)}$	Flow LoI $p_f^{(i,o)}$ or $s_f^{(i,o)}$
L0	00000001
L1	00000011
L2	00000111
L3	00001111
L4	00011111

The federates calculates the Flow LoI of each attribute value update before sending out the update message. This Flow LoI represents the attribute relevance in the update message.

Symbol $\eta_f^{(i,o)}$ denotes the Flow LoI of attribute value update from object instance ***o*** of object class ***i***. Each attribute value update may have different $\eta_f^{(i,o)}$ according to the attributes in the message to update.

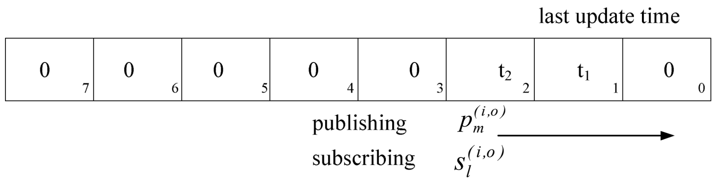


Fig. 5. Flow LoI of Attribute Value Update Message

The Flow LoI $\eta_f^{(i,o)}$ is obtained from the attributes to update values in the update message. As long as there's value of an attribute in some LoI in the update message, the corresponding bit of the LoI is set to 1. The evaluation of $\eta_f^{(i,o)}$ is shown in Table 3. It

Table 3. Flow LoI of Update Message: Evaluation

LoI $\eta_j^{(i,o)}$	Flow LoI $\eta_f^{(i,o)}$
L0	000xxxx1
L1	000xxx10
L2	000xx100
L3	000x1000
L4	00010000

can be seen that any bit lower than $\eta_j^{(i,o)}$ should be zero. Symbol 'x' in Table 3 means both 0 and 1 are possible in the bit.

From the symbol definitions above, we can get the following. On one side, bits of $p_f^{(i,o)}$ represent whether the federate is publishing attributes in the LoI of object instance ***o***. On the other side, bits of $\eta_f^{(i,o)}$ represent whether the attribute value update contains value of attributes in the LoI. So results of bit AND operation on the two Flow LoIs with '&' operator denotes how the attribute value update is matching with the federate's publish. The AND operation of $s_f^{(i,o)}$ and $\eta_f^{(i,o)}$ represents the matching between the update with the subscription, similar to $p_f^{(i,o)}$ and $\eta_f^{(i,o)}$.

Suppose a federate publishes object instance ***o*** in $p_f^{(i,o)}$. When it composes an attribute value update with Flow LoI $\eta_f^{(i,o)}$, RTI can check if the update is in accord with the federate's publish as following.

```
IF( $p_f^{(i,o)}$ & $\eta_f^{(i,o)}$ > 0)
    send out the attribute value update;
```

Suppose the federate subscribes object instance ***o*** in $s_f^{(i,o)}$. When it receives an attribute value update with Flow LoI $\eta_f^{(i,o)}$, RTI can check if the update is in accord with the federate's subscription as following.

```
IF ( $s_f^{(i,o)}$ & $\eta_f^{(i,o)}$ > 0)
    the attribute value update is accepted;
```

To control the object instance update frequency, the federate should provide reference update frequency for each LoI. The value of frequencies depends on bandwidth of the federate. Here we use function *f(loi)* for standard update frequencies of the federate. For each LoI, its minimum update interval is given on the corresponding frequency.

Minimum Update Interval(MUI). $\triangle t(l) = [1000/f(l)]$ (ms)

RTI preserves an array of last update time for each LoI of the object instance as Figure 6. If a new attribute value update is accepted, MUI of LoI bits that are embraced in $\eta_f^{(i,o)}$ will be updated in the array.

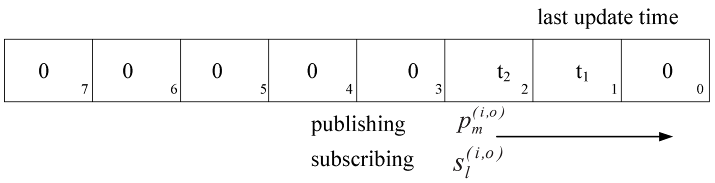


Fig. 6. Last Update Time for Federate Publish/Subscription

So long as MUI of one LoI has timed out, next attribute value update with accordant $\eta_f^{(i,o)}$ will be accepted. The logic is as the following.

```
l = object.loi; // $p_m^{(i,o)}$ for local object instance or
                // $s_l^{(i,o)}$ for remote object instance
t = current_time;
IF( t-t1, t-t2 > $\triangle$t(l))
  accept the update;//for remote object instance.
  //or "send out the update" for local object instance
ELSE
  drop the update;
ENDIF
```

In this way, the bandwidth usage can be cut down and update fidelity of every LoIs is guaranteed. It should be noted that attributes of L0 is subscribed by all subscribers. When an attribute value update brings value of L0 attributes, the bit of L0 is certain to be 1 in the AND operation. So the critical attribute value updates won't be dropped. It will be send out by the sending federate or be accepted by the receiving federate.

5 Experiment Evaluation

In this section, simulation experiments are conducted to evaluate the flow control on low-bandwidth federates. We use four computers to conduct the experiment. The computer setup is shown in Table 4. The sender computers, S1 and S2, are responsible for a BH RTI containing 500 object instances individually. Each receiver computer, R1 or R2, starts a BH RTI receiving all the update messages of total 1000 object instances. To investigate the flow control independently in the receiving computers, flow control is disabled in the sender computers. In this way, we can see what ability the receivers own to treat the traffic. We assume the maximum bandwidth of R1 is 30 Mbps and that of R2 is 10 Mbps. All the incoming packets are looked on as "original data", and packets after flow control are looked on as "data after flow control". Then we can see the flow control efficiency in host R1 and R2.

Table 4. Computer Setup for Experiment

Host Id	CPU	RAM	OS
S1	P4 3.2G	1 G	winXP
S2	P4 3.2G	1 G	winXP
R1	P4 3.0G	512M	winXP
R2	P4 3.0G	512M	winXP

The experiments are carried out in 100M Ethernet network using Huawei Switch Quidway S3050. According to the bandwidth restriction of R1 and R2, *f(l)* is set for them. Here are the settings.

Packet size: 220 bytes, including 10 attribute values, 8 bytes for each.
S1, S2: no flow control. 500 objects in L2
R1: 400 objects in L2 and 600 objects in L1; *f*(L2) = 20, *f*(L1) = 5.
R2: 400 objects in L2 and 600 objects in L1; *f*(L2) = 10, *f*(L1) = 2.

The bandwidth usage results are shown in Figure 7. From the figure we can see that R1 and R2 have endured the total bandwidth more than 70 Mbps. But after flow control, the bandwidth is cut down to about 20 and 10 Mbps respectively. Then we select

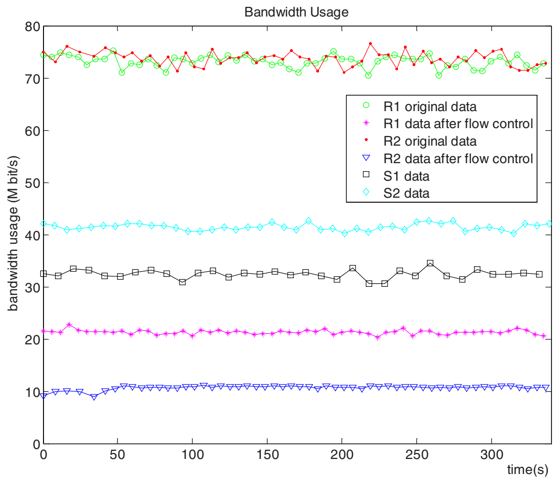
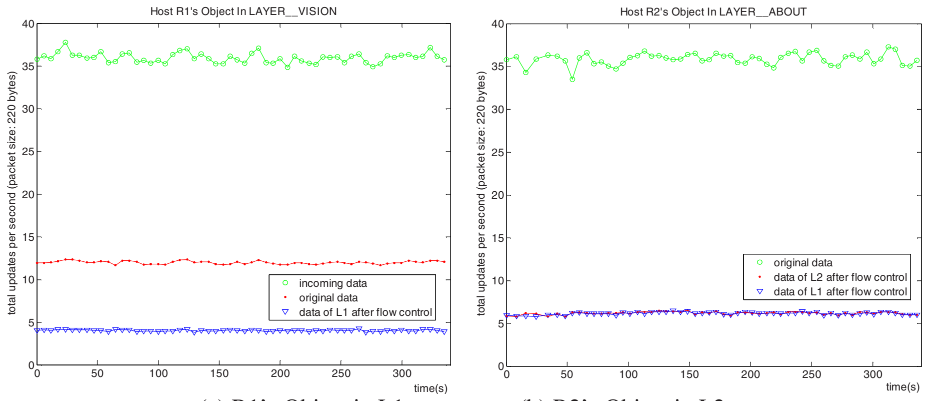


Fig. 7. Bandwidth Usage

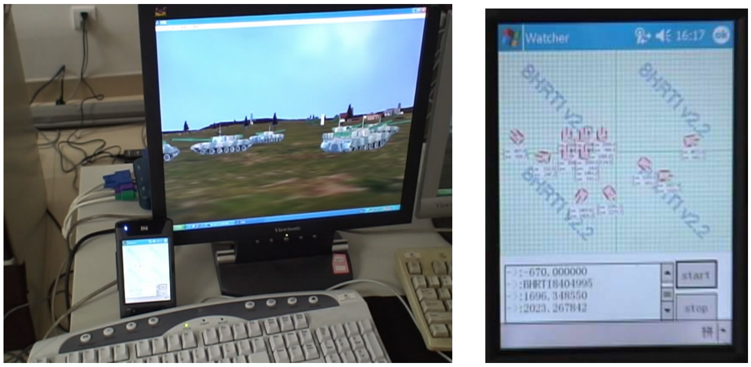


(a) R1's Object in L1 (b) R2's Object in L2

Fig. 8. Object Update Results

two object instances from R1 and R2 to check the update fidelity. From Figure 8 we can see that each object instance maintains a successive update frequency. The total update frequency of object instance in L2 is frequency of L1 plus that of L2 plus that of L0. The update frequencies of object instances of L2 are high enough in each LoI, L1 or L2, for the application to treat with. The update frequencies of object instances of L1 are also successive. The successive update in each LoI guarantees real-time update for each attribute with the help of LoI.

We have implemented the LoI-based flow control in BH_RTI 2.2. The screenshot of an application is shown in Figure 9. Three computers are used to simulate 16 tank entities, including one wireless laptop. A PDA is used for the observer federate, which is based on a wireless connection of the bandwidth about 20 Mbps.



(a) Simulation Scenario (b) Federate on mobile PDA

Fig. 9. Screenshot of the Interaction between wired and Mobile Federates

6 Conclusion

This paper suggests to reduce the object updating frequency and to restrict the update interval based on LoI according to the federate bandwidth. A flow control model on low-bandwidth federates was built on the basis of LoI. This model depicts the way to conduct flow control with fidelity guarantee. Flow control algorithms are proposed to drop update packets according to the object LoI and packet LoI. Experiments illustrated that the flow control model can control the bandwidth of flow on federate bandwidth requirements.

Acknowledgement

This work is supported by the National 863 High-tech program of China under grant number 2006AA01Z331 and National Natural Science Foundation of China under Grant No. 60603084.

References

1. Defense Modeling and Simulation Office, High Level Architecture interface specification version 1.3, April 1998.
2. IEEE standard for modeling and simulation (M&S) High Level Architecture (HLA) – Framework and rules. (IEEE Std 1516-2000). New York:The Institute of Electrical and Electronics Engineers Inc., 2000.
3. IEEE standard for modeling and simulation (M&S) High Level Architecture (HLA) – Federate interface specification (IEEE Std 1516.1-2000). New York:The Institute of Electrical and Electronics Engineers Inc., 2000.
4. IEEE standard for modeling and simulation (M&S) High Level Architecture (HLA) – Object Model Template (OMT) Specification (IEEE Std 1516.2-2000). New York:The Institute of Electrical and Electronics Engineers Inc., 2000.
5. Pullen M, Myjak M, Bouwens C. Limitations of Internet protocol suite for distributed simulation in the large multicast environment. RFC 2502, 1999.
6. S.Bachinsky, J.Noseworthy, F.Hodum, Implementation of the Next Generation RTI, Spring Simulation Interoperability Workshop, Orlando, FL., USA, 1999.
7. Defense Modeling and Simulation Office, Department of Defense. 2002. RTI 1.3 – Next generation programmer's guide version 6. http://www.dmso.mil/.
8. Mikael Karlsson, Lennart Olsson, pRTI 1516- Rationale and Design. Fall Simulation Interoperability Workshop, Orlando, FL., USA, 2001.
9. Douglas D Wood, Len Granowetter. Rationale and Design of the Mak Real-Time RTI. Spring Simulation Interoperability Workshop, Orlando, FL., USA, 2001.
10. J.O. Calvin, C.J. Chiang, S.M. McGarry, S.J. Rak, D.J. Van Hook, M.R. Salisbury, Design, implementation, and performance of the STOW RTI prototype (RTI-s), Proceeding of Spring Simulation Interoperation Workshop, 1997, Paper 97S-SIW-019.
11. Suiping Zhou, Stephen John Turner, Wentong Cai, Hanfeng Zhao, Xiaolin Pang. A Utility Model for Timely State Update in Distributed Wargame Simulations. Proceedings of the 18th workshop on Parallel and Distributed Simulation. pp. 105-111. 2004.
12. BH RTI 2.3 User Guide. http://www.hlarti.com/
13. Zhou Z, Zhao QP. Reducing time cost of distributed run-time infrastructure. the 16th International Conference on Artificial Reality and Telexistence, Hang Zhou, Nov. 2006
14. Zhou Z, Zhao QP. Research on RTI congestion control based on the layer of interest. Journal of Software, 2004,15(1):120~130.
15. Zhou Z, Zhao QP. Extend HLA with layered priority. Proceedings of the Spring Simulation Interoperability Workshop. Orlando FL, 2003. Paper 03S-SIW-012.