A Hybrid Network Using HLA Federation with Conservative Data Compensation

Tainchi Lu*, Guanchi Wu, and Chinghao Hsu
Department of Computer Science and Information Engineering
National Chiayi University
Chiayi, Taiwan, R.O.C.
E-mail: tclu@mail.ncyu.edu.tw

Abstract

In this paper we propose a distributed federation that it comprehensively incorporates with wired and wireless network environments, and IEEE 1516 high level architecture (HLA) to not only support seamless data communication but also facilitate mobile hosts to get the higher interoperabilities. We elaborate the IP-based mobility management agent with forwarding client mechanism and deploy federate agents to achieve a low-complicated but high-practical mobility management for a wide variety of mobile devices. As a conventional wireless LAN is concerned, it inherently suffers from an information suspension and intermittent message loss as long as a phenomenon known as handoff occurs. Accordingly, we devise a conservative data compensation model (DCM), namely backup starting point scheme, to cope with the information suspension problem incurred by the handoff procedure. In particular, the DCM would not aggravate distributed servers' computing loads and exhaust buffer space during client's short-term disconnection phase. As a consequence, the simulation results indicate that the proposed architecture of HLA federation supports the efficient mobility management, seamless data communication, and low system resource consumption for wireless and mobile distributed systems.

Keywords: High Level Architecture, Wireless Local Area Network, Data Compensation, Mobility Management, Distributed System

1. Introduction

Nowadays, demand for Internet technology has been rapidly growing to extend to cooperate with distributed computation, wireless communication, and mobile computing technologies [9]. Consequently, the modern networking usage of a user is not restricted to depend on cable connections and use desktop computers which are placed in a specific computer room. Therefore, the motivation for us to integrate wired/wireless LAN with IEEE 1516 High Level Architecture (HLA) to establish a distributed federation system [5, 6, 8] is to make users more easily to access the ubiquitous high-efficiency network in any time and any where by any device. We take the mobility management of moving clients as an essential functionality and develop three significant components corresponding to the distributed federation, namely federate agent (FA), IP-based mobility management agent (IP-MMA), and data compensation model (DCM). The proposed HLA federation architecture not only incorporates with the wired and wireless local area networks, but it also takes desktop PC, notebook computer, PDA, and cellular phone into account to facilitate the higher interoperabilities among these diverse devices.

In this paper, we propose a data compensation model (DCM) to solve the forementioned problem. When a FA occasionally receives a disconnection signal from a mobile client, it makes use of DCM to backup the messages which are sent from the other mobile clients to this disconnected mobile host. Moreover, we take advantage of HLA time management service to guarantee data-centric consistency and absolute event order by means of generating time-stamp-ordered (TSO) messages. Therefore, we adopt backup starting point (BSP) scheme to cope with data inconsistency problem and increase the FA's buffer utilization when mobile hosts have handoff within WLAN [2, 3, 10, 11].

The rest of this paper is organized as follows: Section 2 gives a brief description towards IEEE 1516 high level architecture (HLA). We first propose our architecture of HLA distributed federation in Section 3. The data compensation model (DCM) with backup starting point scheme is described in Section 4. The experimental results are shown in Section 5. Finally, we conclude the paper and raise the future work in Section 6.

---

[*] Corresponding author

2. Related Work

In recent year, the area of modeling and distributed simulation has been studied intensively by many research agencies, such as the Simulation Interoperability Standards Organization (SISO), MITRE Corporation, and the NPSNET Research Group. In particular, the U.S. Department of Defense (DOD) has driven the evolution of standards to support the interoperability of simulations in order to reduce costs and improve qualities. In this section we briefly review the IEEE 1516 HLA.

High Level Architecture (HLA) [12] is an integrated and general-purpose distributed architecture, which is developed by the DOD to provide a common interoperable and reusable architecture for distributed modeling and simulation. The HLA was approved as an open standard IEEE 1516 through the Institute of Electrical and Electronic Engineers in September 2000. The HLA consists of three core components as shown in Table 1, including federation rules [13], interface specification [14], and the object model template (OMT) [15]. A federate is defined as a member of an HLA federation, so all simulating applications participated in a federation are specified as federates. Hence, an HLA federation is a set of interacting federates to aim at carrying out a specific objective.

The federation rules describe the responsibilities of federates and their mutual relationships with the run-time infrastructure (RTI). The interface specification identifies how federates will interact with the federation through the RTI. The HLA OMT [15, 16] provides a common framework for all objects and interactions managed by a federate. Its components consist of object model identification table, object class structure table, interaction class structure table, attributed table, routing space table, and FOM/SOM lexicon. In other words, all data types for each of these object attributes need to be defined accurately according to the OMT format and syntax. The RTI provides sufficient services for simulation systems and specifies the federate interface using the following languages: CORBA IDL, C, C++, Ada 95, and Java. It is available to support Solaris, IRIX, AIX, HP-UX, Windows NT and Linux operating systems. Moreover, the RTI services are classified into six categories that describe the interface between federates and the RTI. For more detailed description of HLA, you can refer to [13, 14, 15, 17].

Table 1. The three major components of HLA

HLA Components	Descriptions
Federation Rules	There are ten rules. Five related to the federation and the other five related to the federate.
Interface Specification	**Six categories in the RTI:** Federation Management (FM) Declaration Management (DM) Object Management (OM) Ownership Management (OWM) Time Management (TM) Data Distribution Management (DDM)
Object Model Template (OMT)	Federation Object Model (FOM) Simulation Object Model (SOM) Management Object Model (MOM)

3. HLA Distributed Federation

In this section, we propose the architecture of an HLA distributed federation and further describe three crucial components of the architecture: federate agent (FA), IP-based mobility management agent (IP-MMA), and data compensation model (DCM).

3.1 Architecture of Federation

With respect to the official specification of HLA, each participating simulation is referred to as a joining federate f. Likewise, the collection of federates which are interconnecting through the run-time infrastructure (RTI) is referred to as an HLA federation F. In other words, the federation F is a set up federates and its definition is shown in (1).

$$F = \{f_{ij}\}, \quad where \quad i, j \in Z^{+} \qquad (1)$$

We designate several individual sub-networks within a local area network (LAN) as a well-defined network area A_i and further build up specific HLA federates, which are referred to as federate agents (FAs) f_{ij} to provide message exchanging and data buffering services. The FAs are widely spread across the whole specified areas to aggregate an HLA federation F. The relationship among an area A_i, a federate agent f_{ij}, and a federation F is expressed in (2) and (3).

$$A_i = \bigcup_{j=1}^{J_i} f_{ij}, \quad where\ 1 \le i \le I \ \ and \ \ I, J_i \in Z^{+} \qquad (2)$$

$$F = \bigcup_{i=1}^{I} A_i, \quad where \ \ I \in Z^{+} \qquad (3)$$

Furthermore, the process interconnections among FAs are simply achieved by the HLA run-time infrastructure, which provides the six major services for simulators to coordinate the operation and exchange update messages. Within an area A_i, an IP-MMA is established to manage the FAs' working and handle with the connecting routes of client hosts. As long as all FAs initially declare to join in the federation, they first need to have registration at their specified IP-MMA in the identical area. The IP-MMA keeps the registration information of each FA in its FA lookup table in order to perform the underlying dispatching operation regarding to the client hosts.

We illustrate the proposed HLA distributed federation in Fig. 1. Fig. 1 shows one HLA federation and three individual areas which each area comprises one IP-MMA and several FAs. In practice for our experimental environment, we adopt 1000Mbps Gigabit Ethernet to communicate three areas with each other and set up the IEEE 802.11b wireless networks in the range of each area.

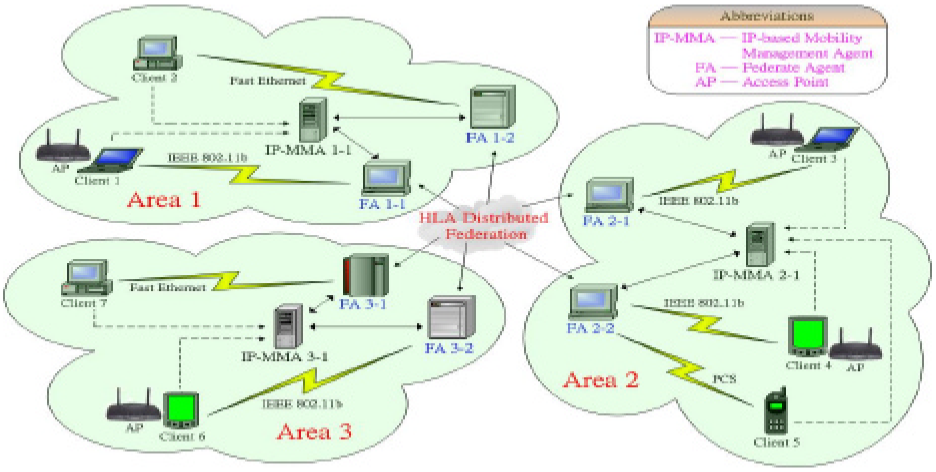

Fig. 1. A demonstrated HLA distributed federation.

3.2 The Underlying Dispatching Operation

In this sub-section, we take a look at the underlying dispatching operation regarding to an IP-based mobility management agent (IP-MMA). The IP-MMA is set up within an independent network area to not only dispatch client hosts to connect with HLA federate agents but also provide substantial mobility services to the moving clients. Fig. 2 shows the connection diagram corresponding to the IP-MMA and a mobile client in a single area. In the following, the six steps are required to accomplish the underlying dispatching operation but except the process of the forwarding client mechanism.

Step 1: In the beginning, an HLA federation is declared to create and start a well-defined distributed simulation, all the active federate agents associated with the federation need to make a registration at their specified IP-MMA.

Step 2: The IP-MMA automatically configures each FA's current connecting circumstance to make sure that every FA is alive with reliable communication.

Step 3: Subsequently, the IP-MMA records the current registration information of the FAs in its local FA lookup table for the sake of executing the client forwarding and dispatching operations.

Step 4: As long as a client intends to participate in the HLA federation, the client can arbitrarily connect to any IP-MMAs that are explicitly belonging to the HLA federation. By means of performing the mechanism of forwarding client, the IP-MMA can forward a mobile client to an appropriate area which is closer to the current geographical location of the client.

Step 5: To consider the case that the IP-MMA and a mobile client collocated in the identical area. The client connects to the specified IP-MMA for the sake of waiting for IP-MMA's dispatching operation. Consequently, the IP-MMA checks the client's present care-of-address against its FA lookup table to find out an appropriate FA. Furthermore, the IP-MMA writes the client's current connecting information to a message log in order to persistently keep track of its up-to-date connecting status.

Step 6: Thereafter, the client is dispatched by the IP-MMA to connect to the specified FA, and then it successfully joins in the HLA federation for participation.

4. Data Compensation Model

4.1 Classification of Disconnection

In this section, we propose a data compensation model (DCM) to deal with packet loss and data inconsistency problems when mobile hosts are in the process of handoff.

In the simulation, the actual handoff time of a mobile client is referred to as t_H, then we use a ceiling function to obtain a reference time $t(C_m)$ to represent that a mobile client C_m has accomplished a handoff process. The equation is shown in (5).

$$t(C_m) = \lceil t_H \rceil \tag{5}$$

According to the distribution of the mobile clients' handoff processes, we calculate a mean handoff time for the whole mobile clients in the simulation. Equation (6) shows mean of the handoff distribution, $\overline{T_H}$, where t is the maximum handoff time in the distribution, x_i represents the accumulated quantity that the handoff time of the mobile clients is i, and n is the total number of mobile clients.

$$\overline{T_H} = \frac{1}{n}\sum_{i=1}^{t} i x_i, where$$
$$n = x_1 + x_2 + \cdots + x_t \quad and \quad t = \underset{1 \le m \le n}{Max}\left[t(C_m)\right] \tag{6}$$

Considering the fact that the finite memory storage in each FA, it is obviously that a shared buffer method is not applicable to reposit interactive messages for a long time. Table 2 shows the classification of disconnection level which is based on the actual handoff time of a mobile client, t_H. Three cases are discussed as follows:

Case 1: If t_H is less than the well-defined handoff time $a\overline{T_H}$, where a is a constant value and $a \ge 1$, then it can be classified as a state of short-term disconnection. We adopt the DCM to handle with the information suspension problem incurred by a handoff procedure.

Case 2: If t_H is between the range of $a\overline{T_H}$ and $b\overline{T_H}$, where both of a and b are constant values and $b > a \ge 1$, then it can be classified as a state of long-term disconnection.

Case 3: In contrast to the two cases mentioned above, if t_H is more than $b\overline{T_H}$, we specify this condition as a fatal disconnection state and terminate the data-exchanging services to the mobile clients.

Table 2. The disconnection levels of mobile hosts

Criteria	**States**	**Operation**	**Examples**
$0 < t_H \le a\overline{T_H}$	Short-term disconnection	DCM	Handoff
$a\overline{T_H} < t_H \le b\overline{T_H}$	Long-term disconnection	Messages are Logging	Enter a trouble spot
$t_H > b\overline{T_H}$	Fatal disconnection	Service is terminated	Roaming

4.2 Conservative Backup Starting Point Scheme

With respect to the operation of the data compensation model (DCM), each FA employs the specified memory buffer to store all of update messages and interactive information after it is activated within the federation. In other words, FAs individually maintain their shared buffers to reposit the messages sent from the other FAs whether are in the same area or not. However, the limited memory capacity often suffers from an overflow problem for lack of adopting a garbage collection method. For that reason, we propose the conservative backup starting point (BSP) scheme to continuously reposit the necessary update messages in a shared buffer and periodically clear the stale data as well as performing the garbage collection algorithm. In order to keep the shared buffer in the range of appropriate size, we take the mean value of the handoff distribution $\overline{T_H}$, message communication time T_{comm}, and task processing time W [7], which are defined in (6), (7), and (8) respectively, into account to estimate a BSP maximum delay factor e for regulating the shared buffer in a FA f_{ij}. The equation is given in (9). In (9), we add up the mean handoff time, message communication time, and task processing time to get a BSP maximum delay factor e.

$$T_{comm} = \ell + \frac{L}{r} \tag{7}$$

$$W_{f_{ij}} = \frac{1}{m_{T(f_{ij})}} \sum_{l \in R_{f_{ij}}} w_{C(l)} \tag{8}$$

$$e_{f_{ij}} = \overline{T_H} + T_{comm} + W_{f_{ij}} \tag{9}$$

Furthermore, we choose a well-defined time interval d and estimate the next grant time q in order to periodically perform the BSP scheme by means of e and q. If the current wallclock time $T_{f_{ij}}$ of the FA f_{ij} is equal to the next grant time q, we take the wallclock time $T_{f_{ij}}$ to subtract the value of the maximum delay factor e to obtain a new backup starting point BSP. The equation is shown in (10).

$$BSP(B_{f_{ij}}) = T_{f_{ij}} - e_{f_{ij}}, \quad where\ 1 \le i \le I,\ 1 \le j \le J_i \tag{10}$$

As shown in Fig. 4, the different backup starting points are calculated in each FA; subsequently the BSP scheme keeps backup messages in the shared buffer $B_{f_{ij}}$ from the BSP point to the current time $T_{f_{ij}}$ and deletes the data which is preceded the BSP point.

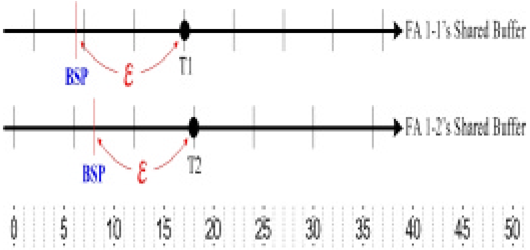

Fig. 4. An example to estimate a new backup starting point BSP for a FA.

As long as the wallclock is allowed to grant to the next one, the shared buffer of the FA is regularly performed the BSP scheme to alleviate the buffer overflowing problem and keep information consistency among the mobile hosts. We give the notation in Table 3 and present the symbols to the backup starting point algorithm. In addition, the pseudo codes of the BSP algorithm are illustrated in Fig. 5.

Table 3. Notation

Symbols	Meaning
I	Number of network areas
J	Number of federate agents in an area
i	Area index, $i = 1,2,\cdots,I$
j	Federate agent index, $j = 1,2,\cdots,J$
$B_{f_{ij}}$	A shared buffer in a federate agent f_{ij}
$e_{f_{ij}}$	A maximum delay factor in a federate agent f_{ij}
$T_{f_{ij}}$	A local time in a federate agent f_{ij}
q	A grant time to perform the BSP scheme
d	A specified well-defined time interval
m_{iact}	An interactive message
m_{rc}	A message to indicate that a client is reconnected
C_m	A mobile client m, $m \in Z^+$
$BSP(B_{f_{ij}})$	The current backup starting point in a shared buffer $B_{f_{ij}}$

```
// Backup Starting Point Algorithm
DCM_BSP ( )
  Create a shared buffer $B_{f_{ij}}$
  Create a local time $T_{f_{ij}}$ to regulate $B_{f_{ij}}$
  while System is Alive
      do switch messages received from the RTI
          case $m_{iact}$
              then $B_{f_{ij}} \leftarrow m_{iact}$
          case $m_{rc}$
              if client's UID is $C_m$ and $C_m$
                  connects to the FA
                then $C_m \leftarrow B_{f_{ij}}$
          if $T_{f_{ij}} = q$
             then BSP ($T_{f_{ij}}$, $e_{f_{ij}}$)
                  $q = q + d$
BSP ($T_{f_{ij}}$, $e_{f_{ij}}$)
   $BSP(B_{f_{ij}}) \leftarrow T_{f_{ij}} - e_{f_{ij}}$
  Keep the data from $BSP(B_{f_{ij}})$ to the current
    time $T_{f_{ij}}$ in $B_{f_{ij}}$ and delete the data pre-
    ceded $BSP(B_{f_{ij}})$
```

Fig. 5. The backup starting point algorithm.

5. Simulation Results

In this section, we concentrate on the efficiency of the proposed data compensation model (DCM) and compare the DCM with a common single buffer method. Moreover, an approximate Poisson distribution $X \sim P(l)$ is adopted to accomplish the discrete simulations throughout the whole experiments. Let X denotes the number of update messages received by a federate agent (FA) in an interval of a time unit. The expected number of receiving messages per time unit for a FA is l, where $l > 0$.

5.1 Buffer Usage and Backup Starting Point Scheme

In this experiment, we first assume that there are not any communication interruptions or network disconnections occurred in the mobile nodes. In addition, we take the buffer size as a major factor to evaluate the backup starting point (BSP) scheme. With respect to the conservative BSP scheme, it relies on the value of a maximum delay factor e to periodically remove the unnecessary interactive messages from the shared buffer. According to the model mentioned in the sub-section 4.2, the value of e has been estimated to be eight in our experimental environment. Moreover, we further take into account the double and half values of the estimated e to compare with each other. The simulation result is shown in Fig. 6. In Fig. 6 we can see that the indexing scheme always keeps the empty buffer usage as the wallclock time is granting. This is because the indexing scheme does not reposit any messages in the shared buffer until an intermittent disconnection of a mobile client is occurred. In comparison with the optimistic indexing scheme, the conservative BSP scheme can keep a stable buffer usage and prevent the shared buffer from occasionally overflowing. Clearly, the factor e plays an important role to dominate the buffer size while the BSP scheme is working to backup the interactive messages for the disconnected mobile clients. The bigger e used in the BSP scheme, the more backup messages stored in the shared buffer. In contrast, we will not store too much backup messages in the buffer when a smaller e value is adopted. However, an inappropriate e will incur insufficient or redundant compensation results in the mobile clients since the BSP scheme cannot make good use of the shared buffer to completely recover the client's lost messages from a data inconsistent state. Regarding to the simulation parameters in the experiments, we suppose that l is equal to 100 messages per second and the size of a packet is about 1500 bytes.

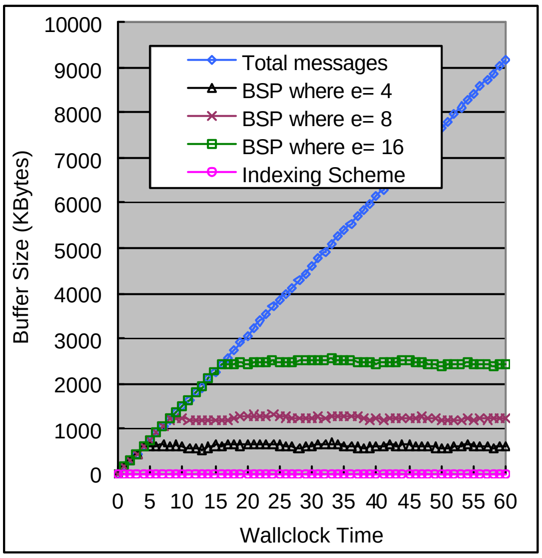

Fig. 6. The DCM by using BSP scheme, where e is 4, 8, and 16 respectively.

In contrast with the preceding experiment, we suppose that a mobile node could move to any network cell with handoff process within an area range. Moreover, the buffer size is also regarded as a crucial factor to compare the conservative BSP scheme with the optimistic indexing scheme and single buffer method. We arrange a client connection schedule in advance to take a look at the variances in the shared buffers which are maintained by different buffering approaches. The schedule is given in Table 4. There are total ten mobile clients simulated in the environment. For instance, the client 1 disconnected from a FA at wallclock time 1 and resumed its network connection at wallclock time 9.

Table 4. The pre-defined client connection schedule

Client Number	**Disconnection Time**		**Client Number**	**Disconnection Time**	
	From	**To**		**From**	**To**
Client 1	1	8	Client 6	25	33
Client 2	3	7	Client 7	26	31
Client 3	6	12	Client 8	32	39
Client 4	19	28	Client 9	50	55
Client 5	21	27	Client 10	52	58

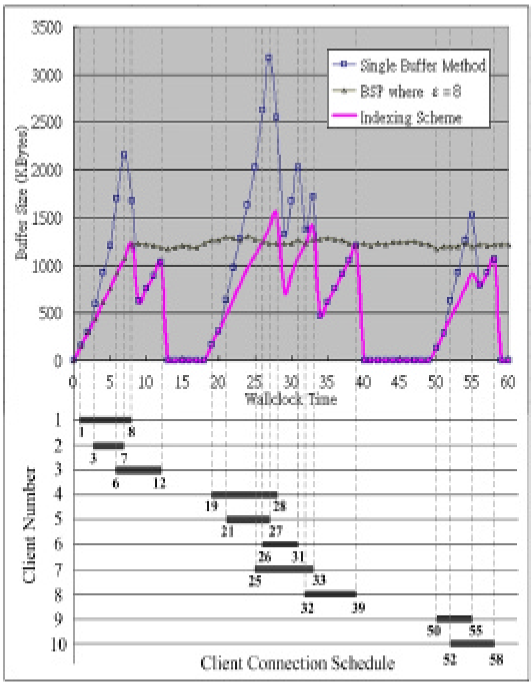

Fig. 7. The buffer variance of the shared buffer associated with the pre-defined client connection schedule.

The simulation result is shown in Fig. 7. In Fig. 7, we can see that both of the single buffer method and indexing scheme reposit backup messages in their shared buffers when an intermittent disconnection occurs to a mobile client. Moreover, concerning to the conventional single buffer method, it creates an individual buffer for each disconnected mobile client; hence, it relatively requires more buffer capacities to support more disconnected mobile clients.

Based on the conservative BSP scheme, we define an equation in (11) to estimate a data compensation rate in percentage concerning to the different handoff times of mobile clients. The data compensation rate of a mobile client C_m is referred to as $R(C_m)$ when the mobile client C_m spent t_H seconds to finish its handoff procedure. In addition, $f(e)$ indicates that the total number of backup messages reposited in a shared buffer as the value e is used to be a BSP maximum delay factor, and $f(t_H)$ means that the total number of interactive messages sent from a FA to the mobile client in the period of t_H seconds.

$$R(C_m) = \frac{f(e)}{f(t_H)} \times 100\% \qquad (11)$$

The simulation result of data compensation rate is given in Fig. 8. The value of e has been estimated to be eight in our experiment. Furthermore, from the distribution of the mobile clients' handoff processes in the experiment, all of the mobile clients accomplish their handoff processes in the range of five to eleven seconds. Hence, the BSP scheme makes a redundant data compensation result when the handoff time of a mobile client is smaller than eight seconds; likewise it gets an insufficient data compensation result as long as the handoff time of a mobile client is more than eight seconds.

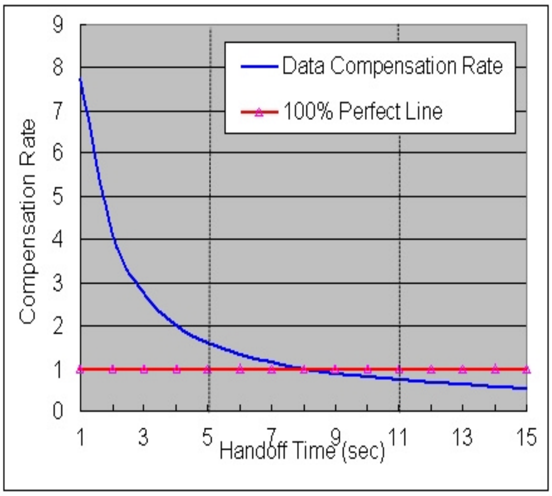

Fig. 8. The change of data compensation rate by using the BSP scheme.

6. Conclusion and Future Work

In this paper, we have proposed the wireless and mobile distributed federation which combines wired/wireless local area network with IEEE 1516 high level architecture to provide client hosts with high-efficient message exchanging services and ubiquitous network access. Particularly, the jointing participants can optionally hold personal communication devices, PDA, notebook computers, or desktop PCs to directly access the network by depending on their current physical network states. The simulation result has revealed that the forwarding client mechanism explicitly works out when mobile clients roam between the two heterogeneous network areas. Moreover, the data compensation model (DCM) has been presented to solve an intermittent data inconsistency problem when mobile clients have handoff across the interior cells within an area. Finally, we have shown that the DCM provides the conservative data compensation service for clients and spend less memory space than the traditional shared buffer method.

Regarding to the future work, instead of adopting the DCM with backup starting point scheme to provide the conservative data compensation, we attempt to further improve the optimistic data compensation model, namely indexing scheme, by making good use of the HLA time management service. Moreover, data distribution management for matching up producers and consumers of data is planned to incorporate with the optimistic data compensation model.

7. Acknowledgments

This work is supported in part by the National Science Council at Taiwan, R.O.C., under grant number NSC91-2213-E-415-004.

8. References

[1] Lawrence Y. Deng, Timothy K. Shih, Ten-Sheng Huang, Yi-Chun Liao, Ying-Hong Wang, and Hui-Huang Hsu, "A Distributed Mobile Agent Framework for Maintaining Persistent Distance Education," Journal of Information Science and Engineering, Vol. 18, pp. 489-506, 2002.

[2] X.Z. Gao, S.J. Ovaska, and A.V. Vasilakos, "Temporal Difference Method-based Multi-Step Ahead Prediction of Long Term Deep Fading in Mobile Networks," Computer Communications, Vol. 25, No. 16, pp. 1477-1486, Oct. 2002.

[3] Zhimei Jiang, Kin K. Leung, Byoung-Jo J. Kim, and Paul Henry, "Seamless Mobility Management Based On Proxy Servers," In Proceedings of IEEE Wireless Communications and Networking Conference, Vol. 2, pp. 563-568, 2002.

[4] A. Kahol, S. Khurana, S. Gupta, and P. Srimani, "An Efficient Cache Maintenance Scheme for Mobile Environment," In Proceedings of International Conference on Distributed Computing Systems, pp. 530-537, 2000.

[5] Tainchi Lu, Chungnan Lee, Ming-tang Lin, and Wenyang Hsia, "Supporting Large-Scale Distributed Simulation Using HLA," ACM Transactions on Modeling and Computer Simulation, Vol. 10, No. 3, pp. 268-294, 2000.

[6] Tainchi Lu, Long-Sheng Li, Yen-Wen Lin, and Guanchi Wu, "A Web-based 3D War-Gaming Simulation built on HLA Distributed Architecture," Journal of Internet Technology , Vol. 3, No. 4, pp. 245-251, Oct. 2002.

[7] Zhiling Lan, Valerie E. Taylor, and Greg L. Bryan, "Dynamic Load Balancing of SAMR Applications on Distributed Systems," Scientific Programming, Vol. 10, No. 4, pp. 319-328, 2002.

[8] Tainchi Lu and Guanchi Wu, "The War-Gaming Training System Based on HLA Distributed Architecture," In Proceedings of the 2002 International Conference on Computers in Education, pp. 889-893, Auckland, New Zealand, Dec. 2002.

[9] J.S. Pascoe, V.S. Sunderam, U. Varshney, and R.J. Loader, "Middleware Enhancements for Metropolitan Area Wireless Internet Access," Future Generation Computer Systems, Vol. 18, pp. 721-735, 2002.

[10] Jon Chiung-Shien Wu, Chieh-Wen Cheng, Nen-Fu Huang, and Gin-Kou Ma, "Intelligent Handoff for Mobile Wireless Internet," ACM Mobile Networks and Applications, Vol. 6, pp. 67-79, January 2001.

[11] I-Wei Wu, Wen-Shiung Chen, Ho-En Liao, and Fongray Frank Young, "A Seamless Handoff Approach of Mobile IP Protocol for Mobile Wireless data Networks," IEEE Transactions on Consumer Electronics, Vol. 48, No. 2, pp. 335-344, May 2002.

[12] U.S. Department of Defense Modeling and Simulation Office (DMSO), *High Level Architecture*. Available through the Internet: http://www.dmso.mil/.

[13] U.S. Department of Defense Modeling and Simulation Office (DMSO), *High Level Architecture Rules, Version 1.3*, Feb. 1998. Available through the Internet: http://www.dmso.mil/.

[14] U.S. Department of Defense Modeling and Simulation Office (DMSO), *High Level Architecture Interface Specification, Version 1.3*, April 1998. Available through the Internet: http://www.dmso.mil/.

[15] U.S. Department of Defense Modeling and Simulation Office (DMSO), *High Level Architecture Object Model Template Specification, Version 1.3*, Feb. 1998. Available through the Internet: http://www.dmso.mil/.

[16] U.S. Department of Defense Modeling and Simulation Office (DMSO), *High Level Architecture (HLA) Object Model Development Tool (OMDT) User's Guide, Version 1.3*, June 1998. Available through the Internet: http://www.dmso.mil/.

[17] U.S. Department of Defense Modeling and Simulation Office (DMSO), *RTI 1.3-Next Generation Programmer's Guide, Version 4*. Available through the Internet: http://www.dmso.mil/