International Journal of Multimedia and Ubiquitous Engineering
Vol. 9, No. 11 (2014), pp. 133-142
http://dx.doi.org/10.14257/ijmue.2014.9.11.13

Improving the Self-adaptive Rate Control to Wireless Multicast for Virtual Environment

Liu Li, Chen Hong-qian, ChenYi and YaoChun-Lian

(Beijing Technology and Business University, Beijing, 100048, China)
{liliu, chenhq, chenyi, yaocl}@th.btbu.edu.cn

Abstract

Distributed virtual environment requires high stability and low delay to support real-time data transmission for large number of users. In wireless network-based virtual environment, traditional transmission protocol fails to meet the requirements of virtual environment due to variety of nodes distribution, complexity, and difference of bandwidth. Based on our research of rate control for distributed virtual environment, we hereby propose a bandwidth self-adaptive rate control algorithm based on layers of interest. According to effective bandwidth of proxy gateway in distributed virtual environment, this method solving the problem of time inconsistency between publishers and subscribers, constraining occupied bandwidth within the range of effective bandwidth of gateways. Our experiments prove that this algorithm can solve the problem that users'needs of interactive with each other very frequently in wireless network virtual environment. The algorithm automatically match data transmission based on bandwidth and the received datagram is smooth and much more stable. The prime performances meet our expected research objectives.

Keywords: *wireless network, virtual environment, rate control, layer of interest*

1. Introduction

With the high growth of Wireless network, distributed interactive simulation must be evolved to adapt to the application on Wireless network. In a large-scale (Distributed Virtual Environment) DVE, hundreds of thousands of entities move and interact in a same virtual environment. While every dynamic object generates status data all the time, the quantity of datagram exchange is massive. Weakening coupling of nodes' interactions by relevant filtering scheme, interest management is one of the most important technologies of large-scale DVE. Under interest management, simulation nodes express their interest range by location of similar features, and therefore only receive data they are interested. However, in the real simulation process, the refresh frequencies are changing all the time and different simulation members have different effective bandwidth. Even adopting interest management, the possibility still exists that entities refresh too fast, exceeding effective bandwidth of simulation members, therefore in turn overloading network and impacting real-time performance and extendibility of simulation. Due to restriction on network, Wireless network also needs data rate control scheme to limit the number of interactive datagram between nodes while ensuring real-time and safe transmission of key data. To solve the dilemma of data transmission in large-scale DVE, in this paper we introduce rate control algorithm based on layer of interest, which can dynamically adjust rate of sending and receiving of status updating datagram according to effective bandwidth of members.

ISSN: 1975-0080 IJMUE

2. Relevant Research

There are many protocols for multicasting in mobile ad hoc networks [3]. But there is no single protocol that jointly addresses QoS and total energy dissipation. [2] Proposed a distributed cross-layer architecture.

In [4], the problem of the minimum-energy information multicast over wireless networks is solved for the single-session case assuming MAI-free operating conditions. [5] Focuses on the joint optimization of end-to-end transport rates, network flows, expected (i.e., long-term averaged) link capacities, and instantaneous (*i.e.*, short-term averaged) power allocation policies in MAI-affected faded coded networks with multiple multicast. Although no convex, theproblemin[5]is proved to optimally solved by dialed composition when the network operates under argotic conditions and the gain of each wireless link is a continuous random variable.

HLA standard provides two interest expression publishing-subscription mechanism: Class-based publishing-subscription mechanism and Value-based publishing-subscription mechanism, which filter data on object class/interactive class level and area data relevance level respectively. However, the common weakness of these two methods the granularity of data relevance is low, without differentiating importance of data. To solve the relevance problem and increase filtering speed and efficiency, zhouzhong[6] raised a relevanceevaluation mechanism: LoI. LoI defines a relevance classification methodology based on the influence of special distances on receiving attributes and attribute values. It defines 6 interest layers based on the interests of receivers: NO_LAYER, LAYER_CRITICAL, LAYER_VISION, LAYER_ABOUT, LAYER_COMPONENTand LAYER_INSIDE, which describes relations of data needs and distances between senders and receivers. The LoI based algorithm in [6] setups different transmission rate for different LoIs based on data needs of nodes, maintains LoI of publishers and subscribers for object entities. And on the ground of LoI matching, nodes control sending and receiving of datagrams on transmission rates according to LoI relevance. This algorithm effectively control data transmission between nodes, but the rate control method may cause inconsistency of attribute value of simulation nodes, and in turn causing inconsistency of member’s attributes.

Because distributed virtual environment (DVE) needs to support more and more users as well as highly increased cooperative works, DVE raised following requirements for data transmission: selective datagram transmission, transmission delay and instability controlling and self-adapting. Traditional specialized transmission protocols are impossible to meet the requirements mentioned above. This paper offers improving LoI-based rate control method to meet the demand of users with various bandwidth in Wireless network environment.

3. Problem Modeling and Analysis

In simulation process, the statuses of entities are changing in real-time, while a new status replacing a previous one. So in general, missing datagram during the status updating process does not affect consistency of simulation. However, if an entity loses updating datagram of while it suspended updating, the status of this entity shows a different status from members of different location, which does cause inconsistency of simulation. The prime objective of the algorithm raised in this paper is to solve the inconsistency of this case. To describe this problem, we define symbols as follows:

LoI: Layer of Interest, representing object class attributes and relevance of attribute values. There are 6 layers, LoI i as l_i;

t_c: Local current time of datagram senders or receivers;

T_i:Timestamp of LoI i, representing receiving/sending/updating progress of local LoI i. On publishing side, T_i is defined as the latest timestamp (the time that datagram was generated) of LoI i. On subscribing side, T_i is defined as the corresponding subscribing time to the latest timestamp that LoI i subscribe to the upper level (in consideration of inconsistency of distributed nodes' clocks);

V_i: Updating rate of LoI i, defined as updating sending/receiving rate of attribute based on the bandwidth demand of publishers/subscribers that publishes and subscribes attribute values;

σ_i: Standard updating period of LoI i, defined as the reciprocal of this layer's standard updating rate: $1/V_i$;

v_i: Real updating rate of Loi i's attributes.

c_i:Datagram buffer for attributes with LoI of i of an object entity

TMR_i: Timing length of a timer for LoI of I;

TS:Timestamp of datagram: the time that datagram was generated by sender;

TS_{ci}: Timestamp of datagram in buffers of LoI of i;

t_c: Retention time of a datagram in buffer;

t_r: Receiving time of datagram: measured by local time of receiver;

In rate control algorithm, we need to adjust entity status and update sending/receiving rate in line with effective bandwidth of members. For instance, to update an object entity's attributes with LoI of i, we need to analyze the generation and sending process of its updating datagram.

In rate control algorithm, data updates by LoI. Because each datagram has many attributes, and the corresponding LoI of these attributes varies. Paper[10] describes LoI of updating datagram as the highest layer of all attributes, therefore the updating datagram may include attributes with LoI of LAYER_ABOUT, LAYER_COMPONENT, and LAYER_INSIDE. So updating multiple attributes might raise the following situations:

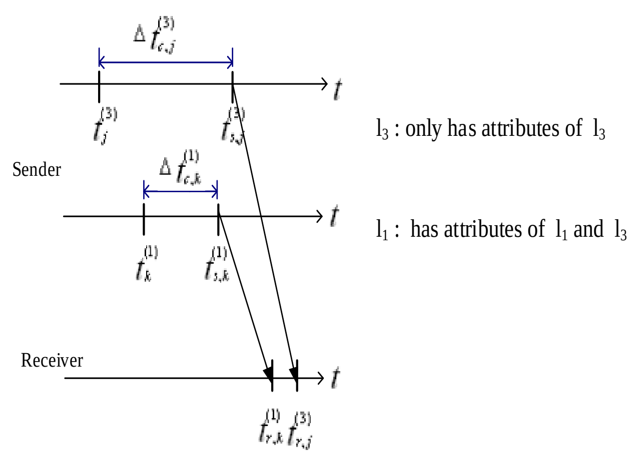

Figure 1. Inconsistency in Status of Attributes

In Figure 1, $t_k^{(1)}$and $t_j^{(3)}$ represent the generation times of a certain datagram with layer 1 and layer 3 respectively;

$t_{s,k}^{(1)}$and$t_{s,j}^{(3)}$ represent the sending times of a certain datagram with layer 1 and layer 3 respectively;

$\Delta t_{c,j}^{(1)}$and$\Delta t_{c,j}^{(3)}$ represent the retention times of a certain datagram with layer 1 and layer 3 respectively;

$t_{r,k}^{(1)}$and$t_{r,k}^{(3)}$represent the receiving times of a certain datagram with layer 1 and layer 3 respectively.

There is only an attribute "a" of layer 3 in the updating datagram of layer3, whereas the updating datagram of layer 1 contains attribute "b" of layer 1 and attribute "a" of layer 3. As one updating datagram may contain attributes of multiple LoI, it could raise inconsistency of attributes status that the earlier generated attributes arrive receivers later than the later generated attributes. Therefore, we must consider the relationships between datagram of different layers in real practice.

4. Rate ControlalgorithmDescription

In the process of DVE, every member registers object class entity, and other members selectively subscribe its attributes set. Rate control algorithm works under the same principle both at the publishers that responsible for updating attributes of entities and subscribers that need to receive updating attributes of entities. We realize rate control algorithm based on our discussion about consistency of attributes statuses and "last datagram" problem mentioned in last section.

4.1. Basic flow of algorithm

In simulation, a member can publish multiple object entities. For attribute sets of various LoIs of each entity, we maintain a time axis and a cache. Every time axis has timestamp T_i, representing last generation time of sending data packet. And we also maintain a timer for each LoI, with timer length TMR_i.

Take an attribute updating process of a random object entity as an instance, the workflow of rate control algorithm is demonstrated as follows:

i. Initial status of simulation, cleaning all caches and starting timers of all LoI

ii. Sending the first attribute updating datagram of entity, recording LoI of updating attributes that the datagram mentioned, and updating timestamps "T_i" of these LoI to the generation time of this datagram.

iii. When the k（k> 1） updating datagram generates, the algorithm check which LoI's attributes were involved. We set the minimum LoI of these j（j <= 6） layers as l_{min}, check the time axis of $LOI_{l_{min}}$, and do the followings: if$t_c - T_{l_{min}} \geq \sigma_{l_{min}}$, send datagram and update all layers $T_j = t_c$; if $t_c - T_{l_{min}} < \sigma_{l_{min}}$, store the datagram to cache of this layer.

When rolling timer of l_i is up, check c_i of l_i and do the followings:

If c_i= NULL, reset the timer; if not, compare $t_c - T_i$ and σ_i, and if $t_c - T_i < \sigma_i$, reset the timer, if $t_c - T_i \geq \sigma_i$y, send attributes of this layer in c_i's datagram, and then in ascending orders continuously test if other attributes of LoIj（j >i） needs to be send. The test method is described as follows:

If $T_j > TS_{c_i}$, then the l_j of this datagram has already been outdated and it needs to be deleted from the datagram. Otherwise, keep the updated attribute of l_j and continuously test until attributes updating of this datagram is over. Record the retention time of it in

c_i as t_c, send the datagram and update timestamps of LoI to generation time of this datagram. And then empty cache c_i and reset the timer.

According to the above steps, the algorithm completes the following functions: updating datagram at standard speed; ensuring "the last" datagram transmit correctly; and making sure the right sequence of attribute layers. The controlling of updating process of receivers is similar, but the time inconsistency between publishers and subscribers needs to be considered.

4.2. Solving the Problem of Time Inconsistency between Publishers and Subscribers

For subscribers of entity attributes, when members receive updating attributes, rate control has the similar basic method to publishers. The only two differences are: when receiving updating datagram, check if LoI of subscribers overlaps with LoI of updating datagram. If not, abandon datagram; If yes, manage datagram sending speed same to the senders do; Due the inconsistency of clocks in distributed nodes, in real practice, the algorithm need to pre-synchronize clocks and take transmission delay into consideration.

In the above steps, publishers needs to compare current time to generation time of last datagram when deciding if need to send datagram. Due to time inconsistency, caused by the difference of physical times between two computers, it might cause errors if receivers use sending time to compare. As shows below:

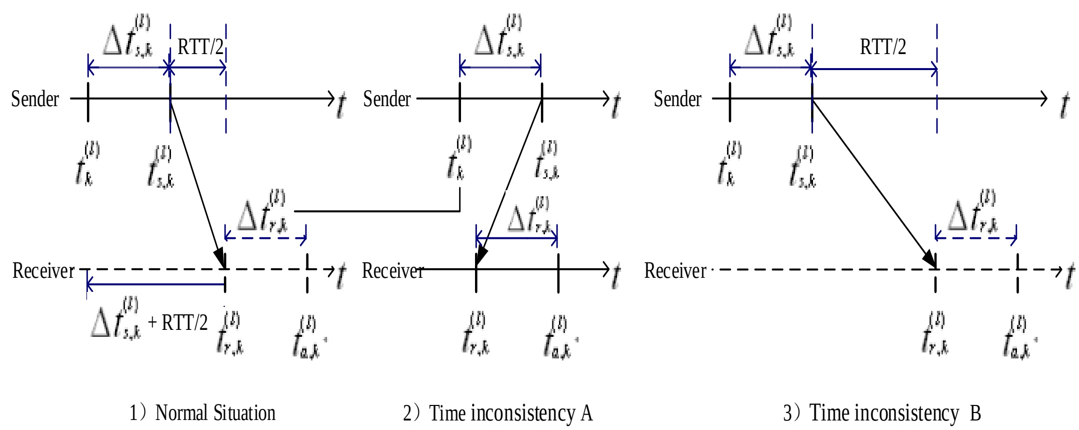

Figure 2. Time Inconsistency between Senders and Receivers

In **Figure 2**, $t_k^{(l)}$ represents the generation time of datagram k, $t_{s,k}^{(l)}$represents the sending time of datagram k, $\Delta t_{s,k}^{(l)}$ represents the caching time in sender's cache of datagram k ($\Delta t_{s,k}^{(l)} = 0$ means sending immediately), $t_{r,k}^{(l)}$ represents the receiving time of datagram k, $t_{a,k}^{(l)}$ represents the submitting time of datagram k, and $\Delta t_{r,k}^{(l)}$ represents the caching time in receiver's cache of datagram k ($\Delta t_{r,k}^{(l)} = 0$ means submitting immediately).

As **Figure 2** 1) shows, in general, datagram receiving time should be greater than sending time ($t_{s,k}^{(l)} < t_{r,k}^{(l)}$). If the physical times of sender and receiver are inconsistent, it may cause 2) and 3). In **Figure 2**2), receiver's time is later than sender's time, which is causing absolute receiving time of datagram earlier than absolute sending time. In**Figure 2** 3) receiver's time is much earlier than sender's time, causing absolute receiving time much later than absolute sending time. Both of these two scenarios should be avoided.

Because on the receiver's side, we also compare current time to the generation time of last datagram (compare $t_{a,k}^{(l)} - t_k^{(l)}$ and σ_l, where σ_l is level l's time interval of receivers), **Figure 2**2) could easily cause $t_{a,k}^{(l)} - t_k^{(l)} < \sigma_l$, , causing abandoning datagram that should have been submitted to the upper layer. And **Figure 2**3) could easily cause $t_{a,k}^{(l)} - t_k^{(l)} \gg \sigma_l$, causing submitting datagram to upper level that should have been abandoned. These two cases will seriously harm the effect of simulation, therefore we should use relative time to compare on receiver's side instead of using absolute time.

Assume the transfer delay of datagram is RTT/2 (RTT is measurable), the time of receiver should be

$$t_{r,k}^{(l)} - \Delta t_{s,k}^{(l)} - RTT/2$$

When sender generates datagram k. As **Figure 2**1) shows, we change the time comparison on receivers to comparison between

$$t_{a,k}^{(l)} \text{and} t_{r,k}^{(l)} - \Delta t_{s,k}^{(l)} - RTT/2$$

Known as the comparison between current time and receiver's time when datagram was generated.

According to the above analysis, in rate control algorithm of receivers, we replace datagram generation time of senders with the corresponding receiver's time when datagram was generated on senders. In rate control algorithm of senders, the value of timestamp T_i represents the generation time of last sent datagram, while T_i in rate control algorithm of receivers, its value represent the receiving time of datagram minus one-way delay $RTT/2$, then minus retention time in cache, also represented as

$$T_i = t_r - RTT\ /\ 2 - t_c$$

5. Efficiency Analysis of Rate Control

To test the impact of this algorithm on data transfer smoothness, we design a test crossing two wireless LANs. The testing environment and data transfer directions are shown as **Figure 3**.

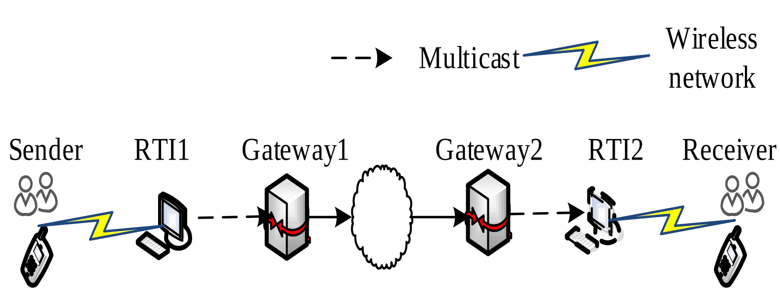

Figure 3. Data Transfer Directions

In **Figure 3**, every LAN has a gateway, a RTI and an application node. To test the impact of this algorithm on data transfer smoothness, we set background sending rate between gateways to 70mbps. And then sender sends a picture with size of 746*483 to receiver based on row sorting of pixels, in order to test receiving situations under rate control and not under control respectively. We evaluate quality of rate control by clarity of the received picture. The size of every datagram is 2238 bytes, which is also the memory size of pixels of one row. Senders sleep 1ms after every 5 datagrams. We set the LoI as LAYER_ABOUT, and the corresponding rate is 150/s, which equals to 1/3 of sending rate.

6. Experiment and Result Analysis

Testing uses bandwidth usages before and after adopting rate control algorithm. We simulate 3 domains in LAN by isolating multicast port, among which, one is responsible for sending while the other two are responsible for receiving. According the design we mentioned in Section 4, the major testing comparison points are the impact of rate control on data transfer smoothness.

Rate control achieves decreasing bandwidth usage by selectively abandon datagram. To some extent, this method relieves traffic jam of network and thereby enables smooth transfer of simulation data. The test is carried out on two domains, one sender and one receiver. Sender registers 1000 entities, with every one of them sends updating datagram with size of 240 bytes repeatedly. We test the transfer smoothness of gateways before and after adopting rate control algorithm. The result is shown as **Figure 4**.

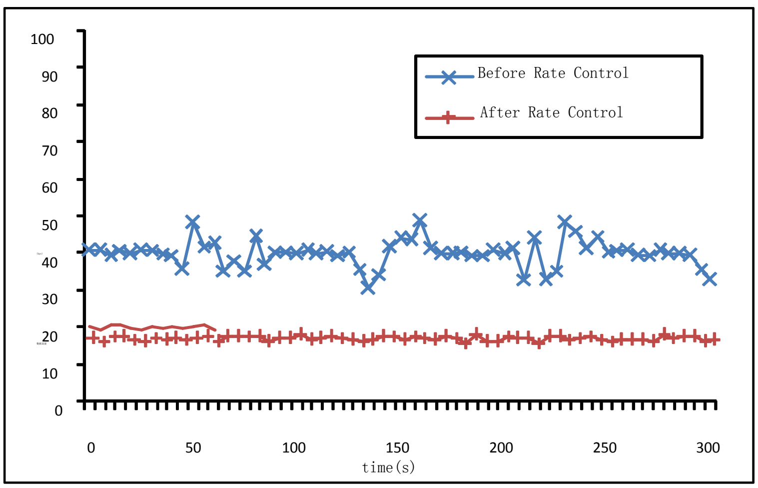

Figure 4. Impact of Rate Control on Data Transfer Smoothness

From **Figure 4**, we observe that the transfer volatility is very high before using rate control algorithm. The reason is that sending rate is too fast for gateway to process, causing a lot of lost datagram. However, after adopting algorithm, the datagram was selectively abandoned. As a result, the received datagram is smooth and much more stable.

7. Conclusion

We analyze the bandwidth features of wireless network environment, and based on these features of variety of nodes distribution, complexity of types, and differences in bandwidth, we cut the data by lower granularity, also known as LoI, and improve the self-adaptive bandwidth rate control algorithm based on LoI. According to its available bandwidth, gateways dynamically send updating information to other gateways in their own communication group on a regular basis. Therefore achieve the purpose of decreasing bandwidth usage and meeting the demands of different available bandwidth of users. The test result shows that the methodcan transfer data according to bandwidth needs of each domain, and the impact on delay is insignificant (<10%). It has met the various needs of virtual environment, therefore we can draw a conclusion that all major performances of this method meet our expected goal.

Acknowledgements

This work is supported by the open funding project of StateKey Laboratory of Virtual Reality Technology and Systems, Beihang University (Grant No.BUAA-VR-14KF-04）and funding Project for Innovation on Science,Technology and Graduate Education in Institutions of Higher Learning Under theJurisdiction of Beijing Municipality(Grant No. PXM2014_014213_000043）. The authors wish to thank the anonymous reviewers for their valuable comments.

References

[1] B. Tavli and W. Heinzelman, “Energy-Efficient Real-Time Multicast routing In Mobile Ad Hoc Networks”, Proc. IEEE Transactions On Computers, vol. 60, no. 5, **(2011)**, pp. 707-722.
[2] M. Van Der Schaar and S. Shankar, “Cross-Layer Wireless Multimedia Transmission: Challenges”, Principles and New Paradigms”, IEEE Wireless Comm., vol. 12, no. 4, **(2005)**, pp. 50-58.
[3] L. Junhai, Y. Danxia, X. Liu and F. Mingyu, “A Survey of Multicast Routing Protocols for Mobile Ad-Hoc Networks”, IEEE Comm. Surveys and Tutorials, vol. 11, no. 1, **(2009)**, pp. 78-91.
[4] Y. Wu, P. A. Chou and S. Y. Kung, “Minimum-energy multicast in mobile ad-hoc networks using network coding”, IEEE Trans. Commun., vol. 53, no. 11, **(2005)**, pp. 1906–1918.
[5] K. Rajawat, N. Gatsis and G. B. Giannakis, “Cross-layer designs in coded wireless fading networks with multicast”, IEEE/ACM Trans. Netw., vol. 19, no. 5, **(2011)**, pp. 1276–1289,.
[6] Z. Zhou and Q. P. Zhao, “Extend HLA with layered priority. Proceedings of the Spring Simulation Interoperability Workshop”, Orlando FL, Paper 03S-SIW-012, **(2003)**.
[7] B. Azzedine and K. Y. Lu, “Optimized Dynamic Grid-Based DDM Protocol for Large-Scale Distributed Simulation Systems”, 19th IEEE Internationa Parallel and Distributed Processing Symposium, USA: IEEE Computer Society, **(2005)**, pp. 1-6
[8] B. Helfinstine, D. Wilbert and M. Torpey, “Experiences with data distribution managementin large-scale federations”, Proceedings of the Fall Simulation Interoperability Workshop, Orlando FL Paper 01F-SIW-032, **(2001)**.
[9] Z. Yang, Y. Yu and B. Klara, “A Multi-stream Adaptation Framework for Bandwidth Management in 3D tele-immersion”, Proceedings of the 2006 international workshop on Network and operating systems support for digital audio and video, USA: ACM, **(2006)**.
[10] Z. Zhou and W. Wu, “LoI: A Filtering and Evaluation Method to the Correlation Data in Distributed Simulation”, Science in China, Scientia F: Information Sciences, **(2009)**, p. 1.

Authors

Liu Li, obtained a PhD degree in computer science and engineering from the Beihang University in 2009. She is a lecturer in the School of Computer and Information Engineering at BeijingTechnology and Business University. Her main research interests are in distributed virtual reality, network.

Chen Hong-qian, received a BEng degree in computer science and technology from Beijing Institute of Technology, China, a PhD degree in computer science and engineering from the Beijing Institute of Technology in 2009. He is an Associate Professor in the School of Computer and Information Engineering at BeijingTechnology and Business University. His main research interests are in virtual reality, Internet of things.

Chen Yi, received a PhD degree in computer science and engineering from the Beijing Institute of Technology. She is a Professor in the School of Computer and Information Engineering at BeijingTechnology and Business University. Her main research interests are in Information visualization and visual analysis.

Yao Chun-lian, obtained a BEng degree in computer science and technology from Beihang University, China, a PhD degree in computer science and engineering from the Beihang University in 2006. She is a lecturer in the School of Computer and Information Engineering at BeijingTechnology and Business University. Her main research interests are in embedded Systems, wireless network.

