Node Selection Algorithm for Peer-to-Peer Network in 3D Virtual Scene

Xiaoying Gu[a,b,c], Yongshan Liu[a,b,c], Xuefang Yan[a,b,c]
[a] Department of Information Science and Engineering, Yanshan University, Qinhuangdao, 066004, China
Email: guxiaoying2006@163.com
[b] The Key Laboratory for Computer Virtual Technology and System Integration of HeBei Province, Yanshan University, Qinhuangdao 066004, China
[c] Data Industry Research Lab, Qinhuangdao, 066004, China

***Abstract*— A new node selection algorithm is proposed to solve the node selection problem for peer-to-peer network in 3D virtual scene. First of all, according to the restrictions of the user's viewing field, the data availability cardinality of the candidate node is larger by setting the priority for the area of interest. Then calculate the effective value through the two performance of node. Finally, user selects the node whose effective value is lager to connect to obtain data. Simulations results show that the algorithm compared to other algorithms, improves effectively the success ratio of requesting data and system throughput, and reduces the based latency for user to obtain data.**

***Index Terms*— 3D virtual scene, peer-to-peer, area of interest, node selection algorithm**

I. INTRODUCTION

IN/recent years, real-time roaming in three-dimensional virtual scene has become a popular trend and been widely used in the virtual campus, virtual exhibitions, large-scale multi-user online games, military simulation and many other areas [1], [2], [3], [4]. Since the three-dimensional virtual scene works in the network to simulate the virtual scene of real-world, the primary task before user begin to roam is building the local virtual scene in client. The current way of downloading the three-dimensional virtual scene is by the network to complete the installation. However, with the dramatic expansion of the scale of virtual scene, the contradiction between the real-time downloading of the massive virtual scene and the limited bandwidth, and the contradiction between the large capacity scenes and the limited storage space for client, makes the transmission of the large-scale virtual scene as the key issues restrict the further development.

In order to solve the above problems, many researchers achieve the fast downloading by introducing the technology of progressive transmission [5] to the three-dimensional network environment. Progressive transmission based on the natural limitations of the user's field of view, only gradually get the scene content which user is interested in without having to download the entire scene.

This work is supported by the Natural Science Foundation of Hebei Province P.R. China under Grant NO.F2013203187, the Innovation Fund for Technology Based Firms of Hebei Province P.R.China under Grant NO.11C1303351006 and the Major Innovation Project for Technology of Hebei Province P.R.China under Grant NO.11213535Z.

We Utilize the real-time transmission to satisfy the user, and it can reduce the amount of the scene data that user requests each time and also can greatly reduce the demand for the local storage space [6]. Existing client-server stores all the contents on the central server and all requests are provided services by the server. Due to the growing number of users with more requests to join in the network, this architecture makes the server overloaded and causes a part of the network congestion, thus constraining the growth of the number of users. The technology of P2P (Peer-to-peer) makes full use of resources, bandwidth and processing power of the node, reduces the load on the central server, and provides support of large-scale users involved in the concurrent system. P2P technology has become the focus of research in recent years [7].

During user real-time working, the basic principle, utilizing P2P to achieve progressive transmission [8] of three-dimensional virtual scene, is as followed: when a user enters and accesses a region for the first time, the area of interest (AOI) and the AOI neighbors in the virtual scene have a certain degree of overlap in the field range. As users reach the area at different time, the firstcomer may have downloaded the scene of the area from the server or the AOI neighbors who had arrived earlier the region. So the latecomers can no longer have to fetch the scene from the server, but download the desired scene by turning to the AOI neighbors. Since the user is constantly moving, its AOI neighbors will also change accordingly. Therefore, how to quickly and efficiently find the AOI neighbors and the available data they have becomes a key issue.

II. RELATED WORKS

Solipsis [9] and VON [10] provided some strategies for finding neighbors, which can notify the new AOI neighbors in the case of not relying on any server with the help of the AOI neighbors or super nodes. After discovering the AOI neighbors, users need to choose the optimal node from many nodes owning different performance and availability data, in order to download data quickly and optimize visual quality [11].

Currently, the relevant literature of node selection is not a lot, aimed to the characteristics of the limits for

doi:10.4304/jsw.9.6.1509-1513

the user's field of vision. In the design of FLOD [12], user firstly obtain the scene description from the AOI neighbors or server through a request list of scene request, when user enters a new unit. Then the user determines objects which is in the visual field and produces a request list of data to obtain the visual objects. By a simple method of query-response, the user firstly send a query request to its AOI neighbors. And the node which owns the data user needs will actively respond to the user. Then the user will randomly select a node from these neighbors to request the data. As all AOI neighbors have an equal opportunity to provide services for users, the strategy of random selection can generally make a certain local load balancing. However, randomly selecting nodes may cause users to ignore the good performance of node and choose the poor ability of node, increasing the possibility of sending the request to the node without service capabilities, result in transmission delay. In addition, when multiple nodes request the same data, the request competition may be caused.

In the literature [13], a strategy of the request of multi-level AOI is proposed for the selection of nodes. The AOI is divided into several concentric layers, wherein each layer closer to the center has a higher request priority. And the nodes in the higher level can be requested in higher priority. If there is more than one node in the same layer, the user will select node randomly. This strategy gets the data requests together closer to the user themselves to avoid the competition among the nodes when several users need the same data. But, when the same layer has more than one node, the strategy will still randomly select node. Moreover, since the user requests the same data from fewer nodes, user may spend more time completing each request, leading to prolong the request time. Some nodes may be the latter to the user's AOI range, then they will not have the data user needs. Thus, they can not provide service for the user when requested data. In addition, nodes have different size of cache in local. For some nodes whose cache is small, nodes will delete the old content when their cache is full. Which makes some nodes is the AOI neighbors of the user, but not own the data that the user need.

Aimed to those problems, an algorithm of node selection based on performance and availability value of the data is proposed in this paper. This algorithm is studied mainly from two aspects: (1) The AOI is divided into two areas by the order according to the data need of user, and setting the priority of accessing to the regional data. By narrowing the range of the user's choice, the algorithm of CA-Value increases the possibility of user choosing the good performance of the nodes to transmission. (2) The algorithm of CA-Value takes into account not only the performance of the node itself, but also the node's deviation distance from the user. The smaller the deviation distance, the greater the AOI overlap range, so the probability that nodes own the data user needs is greater. Finally, according to three evaluation quotas including the rate of request successfully, basic latency and throughput, we can verify that the performance of the algorithm of CA-Value is improved comparing with the existing similar algorithms.

III. The Algorithm Of CA-Value

In order to select the optimal node, the algorithm of CA-Value sequentially computes the service capabilities of node. And the stronger the service capabilities of node, the better the response condition of the candidate. There are two main factors to affect the service capabilities of node, including the performance and the availability of data of nodes. The performance of nodes can be represented by the value of Capacity, it reflects the ability of nodes processing information. And the availability of nodes can be represented by the value of Availability, the value of Availability measures the usability of the node's data for the user. Therefore, the larger the value of Capacity and Availability for each node, the stronger the service ability that the node can provide for users.

A. The values of node's Capacity and Availability

As for any node, there are three factors affecting its Capacity, three factors include the Bandwidth, CPU Speed and Cache. When the Bandwidth is greater, the speed of transmission is faster, then the data can be more quickly transmitted to the local of user. When the CPU Speed is greater, the speed of processing the requests and data is faster. Because each client has a certain cache, the client does not have enough space to store all the scenes with the amount of the scene data gradually increasing. So to accommodate the new scenes, the unwanted scenes need to be removed. When the Cache is greater, more scenes can be stored, and the required data of user has a great probability not to be deleted. The formula of the capacity of the node P is as follows:

$$\begin{aligned} Capacity[P] &= a \times Bandwidth[P] \\ &+ b \times CPUSpeed[P] + c \times Cache[P] \end{aligned} \tag{1}$$

Where $a + b + c = 1$,and $a, b, c > 0$.

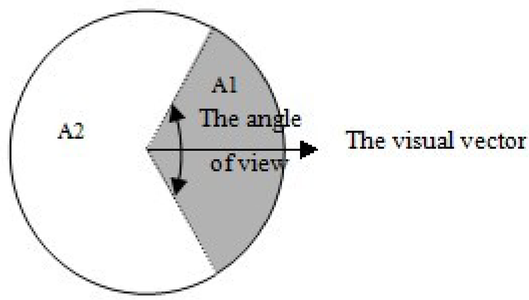

Figure 1. The viewer scope.

When avatar is in stationary, the actual viewable area is only the fan-shaped area in the horizontal plane. And it is only part of the AOI, so the AOI can be divided into two areas $A1$ and $A2$, as shown in Fig. 1. When entering a new unit, the user firstly only can see the objects in the area of $A1$. Therefore $A1$ is called direct viewable area, in which all objects are direct visual objects of users and have the highest priority for transmission. At the same time, $A2$ is a potential viewable area, where the objects are not directly visible by users. But those objects will become directly visible when users move slightly or look around. Therefore once the objects in $A1$ get finished, if there is no new direct visual area, these objects in $A2$ should be transferred and downloaded to the client. These areas are queued and set a priority, namely $A1 > A2$.

According to the priority of the areas, the user should firstly download the related data of $A1$ when entering a new unit. Since the neighbors in $A1$ have more probability of owning the data that user needs than those in $A2$, the user should firstly send request to neighbors in $A1$. Moreover, the neighbor who has the smallest distance and angle deviating from the visual vector of user, has the most probability to have downloaded the data of $A1$, the AOI scope of this neighbor is more similar to the direct viewable area of user and the possibility of that the neighbor node is rejected is larger. So this neighbor node has the maximum probability of having the required data which is needed by user. It is also said that the value of this neighbor's availability is larger.

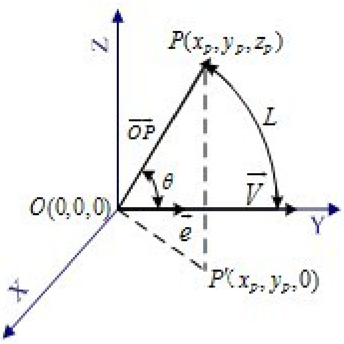

Figure 2. Node coordinates.

In a 3D virtual scene, the user may stop at any height looking down or looking up objects, so they exist in the three-dimensional axes. As Fig.2, assume the coordinates of the user as the origin $O(0, 0, 0)$. The direction of the user's viewable vector in the horizontal plane is Y axis, then the unit vector can be expressed as $\overrightarrow{e} = 0i + 1j + 0k$. For any neighbor P, set its relative coordinates as $P(x_p, y_p, z_p)$, then $\overrightarrow{OP} = x_p i + y_p j + z_p k$. The module length of the vector $\overrightarrow{OP}$ is the radius of AOI, and the angle that $\overrightarrow{OP}$ deviate from the unit vector $\overrightarrow{e}$ is Θ. Then the arc length L which corresponds to the angle Θ is used to represent the distance that node P deviates from the user. So the arc length L is smaller, the availability value of node P is larger.

According to the formula of the arc length, there is: $L = \Theta|\overrightarrow{OP}|$, where $\Theta \in (\pi)$ is the angle between vector $\overrightarrow{OP}$ and unit vector $\overrightarrow{e}$, $|OP|$ is norm of vector $\overrightarrow{OP}$, $|\overrightarrow{OP}| = \sqrt{x_p^2 + y_p^2 + z_p^2}$.

According to the formula of vector scalar product, there is: $\cos\Theta = \frac{\overrightarrow{e}\cdot\overrightarrow{OP}}{|\overrightarrow{e}||\overrightarrow{OP}|}$, then $\Theta = \arccos\frac{\overrightarrow{e}\cdot\overrightarrow{OP}}{|\overrightarrow{e}||\overrightarrow{OP}|}$.

Where $\overrightarrow{e} \cdot \overrightarrow{OP} = 0x_p + 1y_p + 0z_p = y_p$.

By the correspondence between the Availability and arc length L, there is $Availability[P] = \frac{K}{L}$,where K is a constant.

Substituting the above correlation equation, the availability of node P can be calculated as:

$$\begin{aligned} &Availability[P]) = \frac{K}{L} \\ &= \frac{K}{\Theta|\overrightarrow{OP}|} = k\left(\arccos\frac{\overrightarrow{e}\cdot\overrightarrow{OP}}{|\overrightarrow{e}||\overrightarrow{OP}|} * |\overrightarrow{OP}|\right)^{-1} \\ &= \frac{K}{\sqrt{x_p^2+y_p^2+z_p^2}}\left(\arccos\frac{y_p}{\sqrt{x_p^2+y_p^2+z_p^2}}\right)^{-1}. \end{aligned} \tag{2}$$

B. Node's Value

Value represent the capacity of the node's service and it can be expressed as:

$$Value[P] = \alpha \times Capacity[P] + \beta \times Availability[P]. \tag{3}$$

Where α, β respectively represent the weight of the capacity and availability, and $\alpha + \beta = 1$. By several experimental tests, set $\alpha = 0 \cdot 6, \beta = 0 \cdot 4$. The value is stored in each node.

C. The Algorithm Of CA-Value

The algorithm of node selection in this paper can be described as followed:

The algorithm of CA-Value SelectByQuality:

Require: all object O in the AOI of the user;
Ensure: the dataInfo user accesses to data;

```
SelectByQuality(O)
Begin
  flag←0;
  while (NeedData()=true) do //node enters a new unit and need to obtain some data
    Initialize(AOIPart[]);
    startTime();
    for (i=1 to AOIPart [].length) do
      A[]←Peer(AOIPart [i]); //discover nodes in i
      for (j=1 to A[].length) do
        Arr[]←Value(A [i]); //computer the value of each node in the area of i
        sort(Arr[]); //sort by nodes'value in Desc
      end for
      for (j=1 to Arr[].length) do
        if (GetData(Arr[[j])=true ) then
          flag←1;//obtain data from nodes sequentially util complete the data user needs
        end if
        if (flag=1) then
          break;
        else
          j←j+1;
        end if
      end for
```

```
23:         if (flag=1) then
24:            break;
25:         else
26:            i←i+1;//if the nodes in the area i cannot meet the user, the next area is selected
27:         end if
28:      end for
29:      if (flag=0) then //if the nodes in AOI can not meet the user
30:         flag←SearchDataFrom(server);//obtaining the data from server
31:      end if
32:      endTime();
33:      allTime←endTime()-startTime()
34:   end while
35:   return dataInfo;
36: End
```

Firstly set the initial value for the identifier flag that the user obtains data, then judge if the user needs to obtain new data, start to initialize the user's AOI and divided the AOI into two different priority areas, $A1$ and $A2$. The line 4 starts timing. The line 5∼26 shows the circulation for firstly traversing $A1$, and if the nodes in $A1$ can not meet the user, then continues to traverse $A2$. Where in the line 6 stores the nodes that have been traversed in the array. And the line 7∼10 computers the value of each node by the Eq. (1), Eq. (2), Eq. (3), ordering by Descending. The line 11∼20 means the user select firstly the node which has the greatest value to send request. And if the node owns the data user needs, a connection will be established to obtain data. If the user accesses data successfully, the loop will be finished. Otherwise, continue to obtain the data from next node until the loop is finished. The line 21∼25 determines whether the outer loop continue to execute by judging the identifiers flag. The line 27∼29 means if the nodes in AOI can not meet the request of the user, server is turned to request data user needs. The line 30 ends the timing and the line 31 computers the time for user to obtain data. The line 33 finally returns the information that user obtains data, in which flag is true means the user obtains data successfully and flag is false means the user obtains data failed.

IV. Experimental simulation and results

A. Simulation environment settings

The performance of the CA-Value algorithm in this paper is tested by the simulation environment based on FLOD. In the simulated virtual environment, several incarnations are simulated through randomly placing some nodes in the scene. At the beginning of the simulation, all nodes stay in the starting position until each node has the data reaches 80% of their vision at average. So each node can have the initial data for sharing to provide services for other nodes. Then nodes move by a certain speed, and reside randomly in certain points of the space. The concrete simulation parameters are as shown in table I.

TABLE I.
Simulation parameters settings

Parameters	Values
Scene scale(unit)	1000×1000
Unit size(unit)	100×100
AOI-radius(unit)	75
time-step	5000
The number of nodes	50-500(increase by 50)
The number of objects	500
The speed of nodeunit/step	1
Cache sizeKB	600

The experiment uses three performance indicators to measure the effectiveness of the algorithm: the rate of success, the basic latency and the throughput. Among those, the rate of success means the proportion between the amount of the user sends requests which obtain data successfully and the amount of the user sends all requests. Assuming that each object can be divided into a substrate and many refined pieces, the substrate gives the broad contour of an object, and the refinement pieces can restore an object to its original appearance. The basic latency means the duration time between the time of starting to request by the user and the time of the substrate is obtained. Throughput is a node receives an average number of blocks per second. Because the algorithm proposed in this paper is also a strategy of node selection, so the experiment compare from two aspects with the strategy of "Multi-level AOI requirements" proposed by literature [13]. For simplicity, "Multi-level represent the node selection strategy by literature [13] and "CA-Value" represent the node selection strategy by this paper.

B. Experimental results and analysis

Election of simulation nodes are increasing from 0 to 500. At first, we compare the request success rate, as shown in Fig. 3, although the request success rate of Multi-level is increasing trend, it is lower than the algorithm of CA-Value. This is because that the Multi-level is not completely avoiding the competition caused by random selection, and there is still the probability of sending a request to the node without service capability. While the algorithm of CA-Value increases the range of selecting the neighbors which owns the availability data, makes the sucsess rate of the user sending request becoming larger.

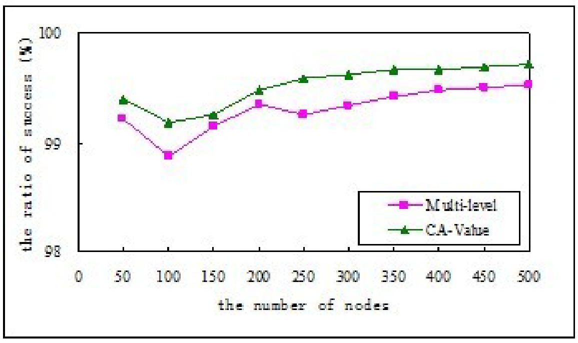

Figure 3. Request success ratio.

Basic latency is an important criterion for evaluating the effectiveness of the algorithm. From Fig. 4 we can find that the algorithm of CA-Value has fewer delay, this is because Multi-level reduces the nodes providing the same number of data and extends the time of completing each request. While CA-Value requests the same amount of data from fewer nodes, but it increases the service capacity of the nodes and maximizes to select the nodes which have good performance. Therefore, CA-Value not only reduces the time waiting for the node to response, but also increases the success rate, then avoids to resend the unnecessary requests.

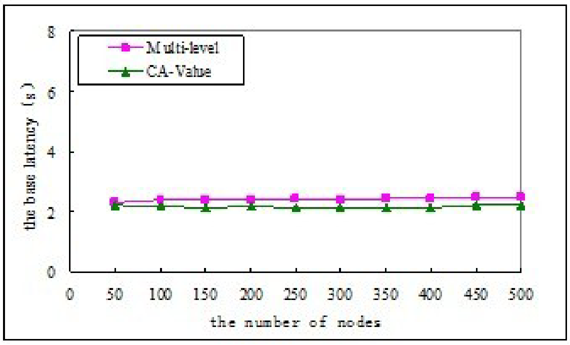

Figure 4. Basic Latency.

With the nodes increasing, we also compare both of throughput, as shown in Fig. 5. The throughput of CA-Value is higher, because this algorithm selects node which is the optimal one and the amount of data exchange is large between nodes. While Multi-level can not guarantee the availability of the data provided by the selected node, and the nodes are switched more frequently, this make the throughput lower.

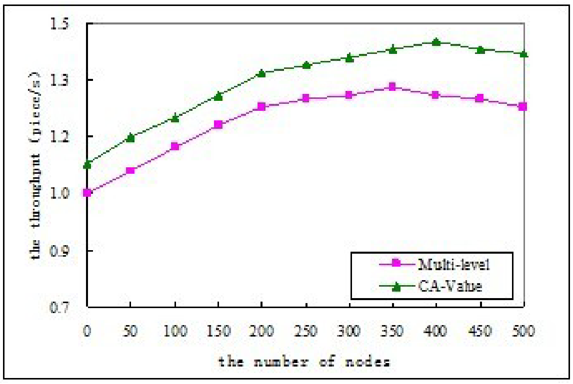

Figure 5. Throughput.

V. Conclusions

In 3D virtual scene based on P2P, it is very crucial for users to browse in real-time. In order to improve the speed of creating the scene, a algorithm of selecting the efficient nodes is needed to access data quickly. Therefore, this paper presents a new algorithm of CA-Value for users to select nodes. This algorithm sets a region priority for users AOI scene and allows users to select nodes in a specific region, reducing the total number of sending requests and narrowing the scope of searching for nodes. Moreover, considering the performance of the node itself, CA-Value also considers the data availability of the node by computing the offset distance of the node and user. So, CA-Value not only improves the success rate of the user requests and system throughput, but also reduces the basic latency. Through the simulation to test and compare, it is proved that CA-Value does improve the effectiveness of selecting nodes.

Acknowledgment

This work is supported by the Natural Science Foundation of Hebei Province P.R. China under Grant NO.F2013203187, the Innovation Fund for Technology Based Firms of Hebei Province P.R.China under Grant NO.11C1303351006 and the Major Innovation Project for Technology of Hebei Province P.R.China under Grant NO.11213535Z.

References

[1] J. Li, "The research and practice of virtual reality technology and the virtual campus," *Journal of Engineering Graphics*, vol. 32, no. 3, 2011.
[2] X. Wu, F. Kang, and W. Huang, "Software simulation of display and control simulation system for aircraft based on virtual reality," *Journal of System Simulation*, vol. 24, no. 6, pp. 1232–1235, 2012.
[3] X. Qu, L. Bai, Z. Hao, and L. Xiao, "Md3 model loading in game," *Journal of Computers*, vol. 7, no. 2, pp. 521–527, 2012.
[4] Y. Tang, H. Gu, and L. Zhou, "Real time virtual view generation for augmented virtuality system," *Journal of Computers*, vol. 6, no. 5, pp. 873–880, 2011.
[5] P. C, P. M, M. B, and A. W, "View- and scale-based progressive transmission of vector data," in *Computational Science and Its Applications-ICCSA 2011. Lecture Notes in Computer Science*, vol. 6783, 2011, pp. 51–62.
[6] Z. Wang, X. Yu, K. Lu, and D. Qing, "Network-driven dynamic scheduling algorithm for large-scale 3d scene," in *Wireless Mobile and Computing, IET International Communication Conference*, 2011, pp. 384–388.
[7] Z. Zhou, Y. Luo, L. Guo, and M. Ji, "A trust evaluation model based on fuzzy theory in p2p networks," *Journal of Computers*, vol. 6, no. 8, pp. 1634–1638, 2011.
[8] M. Almashor and I. Khalil, "Reducing network load in large-scale,peer-to-peer virtual environments with 3d voronoi diagrams," in *IEEE*, 2010.
[9] L. Genovali and L. Ricci, "Aoi-cast strategies for p2p massively multiplayer online games," in *6th IEEE,CCNC*, 2009, pp. 1–5.
[10] S. Hu, J. Chen, and T. Chen, "Von: A scalable peer-to-peer network for virtual environments," in *IEEE Network*, vol. 20, no. 4, 2006, pp. 22–31.
[11] Z. Yiming, L. Xicheng, and Z. Bingqian, "An efficient search algorithm for large-scale p2p systems," *Journal of Software*, vol. 19, no. 6, pp. 1473–1480, 2008.
[12] H. Shunyun, H. Tingjao, and C. Shaochen, "Flod: A framework for peer-to-peer 3d streaming," in *Proceedings of IEEE INFOCOM. Phoenix, AZ, USA:IEEE*, 2008, pp. 2047–2055.
[13] W. Sung, S. Hu, and J. Jiang, "Selection strategies for peer-to-peer 3d streaming," in *Proceedings of the 18th International Workshop on NOSSDAV. Braunschweig, Germany*, 2008.