International Journal of Future Generation Communication and Networking
Vol. 8, No. 3 (2015), pp. 191-206
http://dx.doi.org/10.14257/ijfgcn.2015.8.3.18

# F-LBQA: Fair Load Balancing QOS Algorithm in Overlay Network

ZHOU Xiao-Yan[1] and Zhao Kai[2]

[1]*Information Engineering Department of Karamay Vocational and Technical College, Karamay Xinjiang, China*
[2]*Teaching Affairs Division, Pingdingshan University, Henan Pingdingshan, 467000, China*
*karamayzxy@126.com*

### Abstract

*Due to the traditional overlay network QOS routing algorithm fails to consider the effects of the resource bottleneck on the routing selection, it results in the local congestion of the system. Therefore, this paper proposes a fair load-balancing QOS overlay routing F-LBQAR algorithm. The algorithm introduces the resource fairness index, sets up a new load-balancing utility function, and adopts the adaptive weighted method in the system's load state to correct the weight restriction of QOS. The experimental results show that the algorithm in this paper can reduce the probability of network congestion caused by resources bottleneck and improve the QOS service success rate and the throughput of the system.*

***Keywords:*** *Cost Model, Service Resources, Routing Selection, Covering Space*

## 1. Introduction

Service overlay network (SON) is a virtual network built on the basis of the basic network, which is run or managed by he third party, and makes a profit through providing high quality data value-added services to the users. In recent years, the research and development of SON has raised concerns from many researchers and Internet service providers. However, the selfish routing policy implemented by the overlay layer of SON tends to cause the imbalance of the flow of the overlay layer, which makes the network resources can not be reasonable used.

The overlay network QoS routing is a very important research direction in the overlay network technology, the overlay network is also called the overlapping network, which refers to the virtual application-layer network constituted by establishing the intelligent nodes in the key positions of the basic network and connecting the nodes with the single cast routing way [1]. This kind of network lies between the user terminal and the physical network, which can be used to implement specific applications. Relative to the underlying infrastructure network, the overlay network can monitor the status information in the underlying network, such as routing trend, transmission delay, flow distribution, congestion status, as well as abnormal failure and so on. On the one hand, rely on the overlay node the overlay network layer interchanges the information, and on the other hand, it based on the intelligent control and management measures of these information, such as the choice of routing detection, flow control, content retrieval, classification as well as management and so on. The advantages of the overlay network are that it effectively improves the quality of service of the distributed network application without changing the architecture of the basic network [2-6].

Because the traditional TCP/IP network can only provide the best-effort service model, which cannot effectively guarantee the transmission quality of the distributed application. While by rebuilding the reliable overlay routing network in the application layer can

ISSN: 2233-7857 IJFGCN

provide the transmission control mechanism with QoS guarantee, this technique is called the overlay network QoS routing. In this architecture, the overlay network is made up of server host nodes in the application layer [7-9]. The nodes can not only transfer and forward data flow, but also have special computing power and storage capacity, and they realize the complex service composition through collaboration with each other. Thus the overlay network can use to routing selection method to effectively solve the resource scheduling of the distributed application.

The resource scheduling of load-balancing is the key problem in studying the application of the distributed system. Because of the differences between the network environments, the volatility of the business throughput in different regions and the unreasonable resource allocation strategy will both result in the imbalance of the network load, so as to cause the congestion phenomenon of the network and reduce the overall throughput of the system. At present, there is still not a good solution to he problem on the IP network. While the overlay network QoS routing technology can provide a method based on the deployment load balancing resource scheduling strategy of the application layer, which helps the distributed system to realize the fair distribution of resources, thus it becomes a current research hotspot [10-12].

The classic overlay network load-balancing QoS routing algorithms mainly include QUEST, PBLCP, RBLCP etc, the basic principles of these algorithms are deriving the representation of the overlay network transmission, calculating the resources consumption cost and the heuristic utility function, based on the shortest path algorithm to search the optimal load-balancing service path, which is helpful to choose the path with overall less consumption of resources [13-14]. However, this method still has a problem that it fails to consider the problem of resource bottleneck caused by the links with a local heavy load. The bottleneck will lead to the congestion of the system, and even will reduce the system throughput and the load-balancing degree when it is serious. Aiming at the problem, this paper focuses on studying the method for calculating the path bottleneck degree and introduces the method into the traditional routing model, then proposes a fairer load-balancing overlay routing algorithm (F-LBQA). The experimental results show that this algorithm can further improve the throughput and the load-balancing degree of the network system.

## 2. The F-LBQAR Algorithm

*A.Coverage Analysis*

In the practical application of the wireless sensor network whose nodes are randomly deployed, the achievement of keeping complete coverage to the monitoring area constantly is not only difficult, but also time-consuming, while it is feasible and acceptable to keep coverage to most of the regions in a certain period of time. As long as the working nodes can maintain a reasonable coverage ratio to the monitoring area, most of the applications can be satisfied, therefore, the coverage ratio can be used to measure the service quality of the network.

The goal of the overlay network load-balancing QoS routing is to find a suitable path for the service composition in the overlay network, which has a relatively small resource cost when accessing business and can make full use of the system’s "free" resources. So in the process of continuous access to the user’s requests, the load can be evenly distributed within the inside of the network space, so as to achieve the maximum of the utilization of the network resources. The types of resources in overlay network can be abstracted as two types, the transmission resources and the computing resources, which can generally be expressed by the transmission bandwidth of the links (BR) and the node's computing power (CR) [15]. The goals of all the current load-balancing QoS routing algorithms are to optimize the distribution of these two kinds of resources and solve the optimal QoS service path. QUEST is the earlier proposed overlay routing algorithm with multiple QoS optimization targets, which proves that this kind of problem is a problem of NP2Complete. The algorithm firstly defines the cost function of multiple QoS targets optimization, and carries out the weighted adjustment to the cost function according to the degree of focus on the problem target, then uses the shortest route $D_{ijkstra}$ algorithm to solve the optimal routing results. Aiming at the load-balancing target optimization, QUEST adopts two kinds of constraint items for QoS metric.

(a) The bandwidth consumption rate of link $j_k$ , $k=\left(t_x,t_y\right)$ is $h_z\left(j_k,h_z t\right)$, among which $h_t$ is the bandwidth resources requirement constraint of the service request.

(b) The computing resources consumption rate of node $h_z$ is $h_a\left(j_k,x,b_t\right)$, among which Ct is the computing resources requirement constraint of the service request.

Set $bt_j$ and $bh_{z,x}$ to the available resources of node j and link $t_{x,y}$, the obtained link heuristic cost function $k\left(t_{x,y}\right)$ is as follows:

$$\begin{aligned} b\left(t_{y,x}\right) &= h_z\left(j_{k,x},b_t\right)+h_a\left(j_k,x,b_t\right) \\ &= b_i / bt_j + b_t / h_{z,x} \end{aligned} \tag{1}$$

The price of path $t$ is calculated by the sum of the link's prices, and the overall consumption cost function of path P is defined as follow:

$$Cost\left(t\right)=\sum_{t_{x,y}\in t} h\left(t_{x,y}\right)=\sum_{t_{x,y}\in t}\left[\frac{h_z}{bt_j}+\frac{b_t}{br_{xy\leq}}\right] \tag{2}$$

In 2005, $t_i$ and others put forward a kind of load-balancing routing algorithm PBSP (Proportional Bandwidth Shortest Path) based on transmission resources and the residual ratio of computing resources, which can have a more accurate utilization degree of the characterization resources. The algorithm switches to use the proportion of residual available resources between the node and the link as the option weight, and generates the cost function multiplication mixed measurement, while the price of the path can still be calculated by the sum of all the link's prices, and its cost function Cost (P) is as follows:

$$Cost\left(t\right)=\sum_{t_{x,y}\in t}\frac{th_z}{bt_j-t_i}\times\frac{br_{x,y}}{br_{xy}-b_t} \tag{3}$$

In 2007, Yingbidi and others proposed an improved routing algorithm RBLCP on the basis of PBSP. The algorithm introduces the constraint factors of the global service resources distribution, which can carry out weighting dynamically on the bandwidth and the computing resources cost function based on the overall resources distribution. Thus the algorithm can have the adaptive load balancing according to the bottleneck status of the two types of resources, in which the value of α is the weighting factor, and its cost function Cost (t) is as follows:

$$Cost\left(t\right)=\sum_{t_{x,y}\in t}\frac{1}{\alpha(bt_j+t_i)+\left(1-\alpha\right)\times br_{x,y}b_t} \tag{4}$$

Definition 1: partial coverage

Set the sensing area of sensor node i to Si, if there is a subset of u in all the nodes deployed within the network area $P$, then u is treated as a full coverage to the network area $P$, which makes:

$$P-\bigcup_{i\in u}S_i\cap\neq\Omega$$

Then u is treated as a partial coverage to the network area C.

Definition 2: quality of service

Set the ratio between the area covered by the set u of working nodes with the entire network area $P$ to $R$, namely:

$$R=\left\|\bigcup_{i\in u}S_i\cap P\right\|/\left\|C\right\|$$

$R$ is treated as the quality of service of the network area C.

Definition 3: overlay set

Set the minimum node set in the network area $P$ which meets the quality of service $R$ to $u_R$, and nQ is the number of nodes of $u_R$, then $u_R$ is treated as a $R$ overlay set of C, and nQ is the $R$ minimum covering number of $P$.

Definition 4: ring domain and ring domain coordinate

In the concentric circle area which is centered on the Sink node, the nodes can receive the wireless signals with the same level of transmission power. According to the similarities and differences of the signals received by the nodes, the network can be divided into several ring domains. For each ring domain divided by different powers, from inside to outside, it is respectively marked with a non-negative integer $L(0,1,2,3...)$. If the minimum covering number within its domain is $n_{RL}$, then the non-negative integers $P(L,n_{RL})$ ρ(L, nQL) is treated as the ring domain coordinate of the network, and $L$ and $n_{RL}$ are treated as the location component and covering component of ρ. The rotation cluster formed by the solid and hollow nodes is an overlay classification of the ring domain, and it is shown in figure 1 and figure 2.

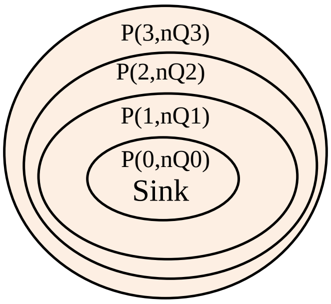



**Figure 1. The Ring Domain Coordinate**

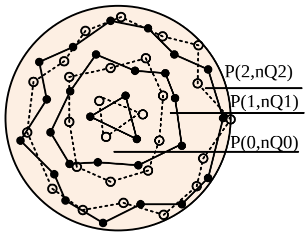



**Figure 2. The Rotation Cluster**

Figure 3 depicts the receiving success rate of the service request in the process when the volume of business constant rising. The results show that when the business volume of the system constantly rising until to the overload situation, the success rate of the service accordingly drops sharply, there is an overload point when the number of the sessions is 80 units per second. This phenomenon proves that the resource bottleneck problem caused by the network congestion will seriously reduce the service throughput of the system.

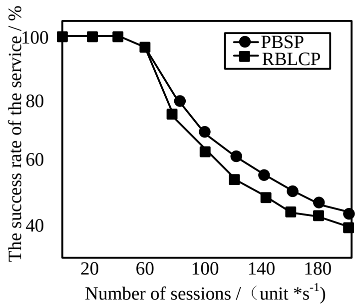



**Figure 3. The Unceasingly Reducing of the Service Success Rate**

*B. The Fair Load-balancing QoS Routing Algorithm (F-LBQA)*

Because of the theoretical analysis and experimental verification in section 2.1, it can be known that the traditional algorithm cannot exclude the local path bottleneck problem. Aiming at that problem this paper proposes a fair load-balancing QoS routing algorithm (F-LBQA), which introduces the constraint factor of the path resource bottleneck degree in the routing model, tries to avoid choosing the bottleneck links, thus can reduce the occurrences of the network congestion to a certain extent. The technical route of the specific algorithm will be given in this section.

The Cost Model of the Service Resource Consumption

The traditional algorithm calculates the service resource consumption by calculating the sum of the link's resource consumption. F-LBQA still uses the similar model, but the difference is that it uses the consumption cost weight of the path instead of the consumption cost weight of the link. This method separately considers the constraint factor of the transmission resources and the computing resources, which is more in line with the physical meaning of the cost characteristics, and is advantageous to carry out the independent weighting according to the different resource QoS requirements.

This section first defines the calculation model of the resource consumption cost weight after the path access the service. The model is divided into two parts, one is the transmission service cost, which is presented by the sum of the link's flow consumption, and the other is the computing service cost, which is presented by the sum of node's computing power consumption. The cost functions are defined as follows.

Definition 1 shows the cost function of the path transmission resources consumption.

Set an arrived business request for $t$, and the QoS requirements of its transmission resources for Bt. Assume that the overlay network system can provide the service request to let the service path P accept the business $t$, then the transmission resources consumption cost of $z$ to $t$ is $t\Delta h(t)$, and it is as follows:

$$\Delta h(t) = h_i / hu_{0,1} + \lambda_i h_t / hu_{1,2} + \lambda_1 \lambda_2 h_t / hu_{2,3} + ... + \prod_{i=0}^{t_p-1} \lambda_j h_{t_p-1} = h_t . \sum_{x=1}^{t_p} \left[ \prod_{j=0}^{i-1} \lambda_y / hu_{x-1,y} \right] \quad (5)$$

$hu_{x,y}$ is the available bandwidth resources stock to jump every ink $t_{x,y}$ on path $t$, $\lambda_1, ...\lambda_i$ is the influence coefficient of the flow for path $t$ to jump every node, the output flow of the node is $Bout = \lambda \times Bin$, $\lambda_0 = 1\lambda$, and $h_t$ is the hop count of the current path.

Definition 2 shows the cost function of the path computing resources consumption.

Set an arrived business request for $t$, and the QoS requirements of its transmission resources for $h_t$. Assume that the overlay network system can provide the service request to let the service path P accept the business $t$, then the transmission resources consumption cost of $t$ to $t$ is $t\Delta h(t)$, and it is as follows:

$$\Delta h(t) = h_1 / hu_0 + h_1 / hu_1 + h_1 / hu_2 + ...h_1 / hu_{t_p} = \sum_{i=0}^{t_p} \frac{h_1}{hu_i} \quad (6)$$

$hu_0, hu_1, ...hu_{t_p}$ is the available computing resources stock to jump every node $i$ on path $t$, which including the original node and the destination node, and , $t_p$ is the hop count of the current path.

It can be known from the above definitions that the two cost constraint items $\Delta h(t)$ and $\Delta u(t)$ are the parameter factors independent of each other. Thus the overall cost function of the path can be calculated through the additive mixing measure, and it is defined as follows:

Definition 3 shows the cost function of the path resources consumption.

$$Cost(t) = \Delta h(t)\,\Delta u(t) \quad (7)$$

Plug formula (6) and (7) into formula (5) can obtain formula (8).

$$Cost(t) = b_t . \sum_{i=1}^{t_p} \left[ \prod_{x=0}^{i=1} \lambda_i / ht_{i-1,i} \right] + \sum_{i=0}^{t_p} \frac{h_t}{ht_i} \quad (8)$$

The Fairness Index of the Service Resource Utilization Rate

In order to reduce the bottleneck degree of the service path, this paper introduces a kind of weights constraint routing method based on the fairness rate of the service resource utilization rate. The method measures whether there is a bottleneck in the path by computing the resource utilization rate of all the links and nodes in the service path. Assume that there is a sequence $y_1, y_2, ...y_m$, and then the fair exponential function $d(y)$ that measures the equilibrium degree of the sequence is defined as follows:

$$d(y) = \frac{(y_1 + y_2 + ... + y_m)^2}{m.(y_1^2 + ... + y_m^2)} = \frac{\left[ \sum_{i=1}^{m} y_i \right]^2}{m. \sum_{i=1}^{m} y_i^2} \quad (9)$$

It can be known from the principle analysis that when there is $y_1 = y_2, ... = y_m$, the value of the fair index $d(y)$ is the greatest value 1, and if the volatility of the difference between sequence $t_i$ is higher, the value of $d(y)$ is smaller, infinitely approach to zero. Thus the domain of $d(y)$ is (0, 1]. If use the mean square error of the numerical sequence to calculate the equilibrium, then the situation is on the contrary, the higher the value is, the greater the volatility will be, the domain is [0,1]. But its value is likely to be zero, which is not good for a division coefficient, so it is more ideal to adopt the fairness index to calculate the equilibrium.

It can be known from the above definition that it is possible to evaluate the load balancing degree of the path through calculating the resource utilization fair index of all the nodes and links on the service path. Assume that the characterization utilization of the resource saturation is adopted, if the saturation degree of the partial links and nodes on the path is relatively higher, which means that the load of the links and the nodes is relatively bigger and will cause the index value be big. If the saturation degree is balanced, the index value will be small, the same it is under the high load. Thus it can reduce the occurrence probability of bottlenecks to treat the index as the regular constraint item of the shortest path weight. The specific calculation method is as follows:

Definition 4 shows the utilization fair index of the path transmission resources.

Set an arrived business request for T, and the QoS requirements of its transmission resources for Bt. Assume that the overlay network system can provide the service request to let the service path P accept the business T, then the fair index of the transmission resources after $p$ access $h$ is $hu(t)$, and its calculation is as follows:

$$hu(t)=\left[\sum_{i=1}^{t_p} b\left(t_{i,j+1}\right)\right]^2 / t_p . \sum_{i=1}^{t_p} b\left(t_{i,i+1}\right)^2 \quad (10)$$

$b\left(t_{i,i+1}\right)$ is the load saturation function of the transmission resources after the link $t_i$, $i+1$ access the request on path $p$, $ht_i$, $i+1$ is the available bandwidth resource stock of the link $t_i$, $i+1$ on path $p$, $br_{t_{i,i+1}}^{total}$ is the total number of the available bandwidth resources of the link $t_i$, $i+1$ on path $p$, and $t_i$ is the hop count of the current path. There is an inverse linear relationship between the index and the transmission load fluctuation of the link, when the volatility degree is relatively higher, and the value of the index will be smaller, which means that the probability of having bottleneck links in the path is greater.

Definition 5 shows the utilization fair index of the path computing resources.

Set an arrived business request for T, and the QoS requirements of its transmission resources for Ct. Assume that the overlay network system can provide the service request to let the service path $p$ accept the business T, then the fair index of the computing resources after $p$ access $h$ is $hu(t)$, and its calculation is as follows:

$$hu(t)=\left[\sum_{i==0}^{t_p} h\left(t_i\right)\right]^2 \left(t_p+1\right) . \sum_{i=0}^{t_p} h\left(t_i\right)^2 \quad (11)$$

$h\left(t_i\right)$ is the computing resources saturation function when hopping every $h\left(t_i\right)$ nodes, including the original nodes and destination nodes, on path $p$, $hu_i$ is the available bandwidth resource stock of the node $t_i$, and $br_{t_i}^{total}$ is the total available computing resources of the node $t_i$. There is an inverse linear relationship between the index and computing load fluctuation of the node, when the volatility degree is relatively higher, and the value of the index will be smaller, which means that the probability of having bottleneck links in the path is greater.

Simply use the index for route choice can only select the paths with high balance degree, but cannot reduce the overall consumption cost of the path. Thus plug formula (11) and (12) into formula (7), add the bottleneck restriction factor on the basis of the service cost model, and calculate the load –balancing degree of the service path through the multiplicative mixed measurement, then the service path which has a small probability of bottleneck and low resource consumption can be obtained. The calculation method of the load-balancing utility function is as follows.

Definition 6 shows the resource load-balancing utility function of the service path.

$$q(t)=\frac{\Delta h(t)}{hu(t)} \quad \frac{\Delta u(t)}{h_c(t)} \quad (12)$$

The Load-balancing Heuristic Function Based on Load Weighted

Formula (12) describes the influences of the service request after its access on the load balance state of the system and the service path, thus the routing selection can be determined according to the weight of the influence degree. It can be known through analyzing the formula that the $\Delta h(t)/hu(t)$ factors are beneficial to improve the load balancing degree of the transmission resources, $\Delta u(t)/h_c(t)$ are advantageous to improve the load balancing degree of the computing resources, and the influence weights of them are completely equal. But in reality, the load state and bottleneck degree of different types of resources during the specific operation cycle of the overlay network are often not the same. Clearly, if the bottleneck situation or load of certain type of resource is heavier, it should be more inclined to optimize the distribution of the resource. Therefore, this paper considers adopting the load distribution of the system to carry out the weighting process of the two kinds of factors, so as to correct the accuracy of the model.

1) The evaluation method of the system's load conditions.

Considering that the goal of the F-LBQAR algorithm is to reduce the fluctuation of the system's resource saturation and improve the balance degree of the overall network's load distribution as far as possible. Thus, this paper adopts the weighting method based on the overall resource fair index, and through calculating the fluctuation degree of the system's overall transmission and computing resources to get the weight value. Formula (13) and (14) can calculate the fair index $R_c$ of the overall computing resource saturation and the fair index $R_c$ of the transmission resource saturation, and this two QoS indexes can carry out the quantitative evaluation on the load balancing degree of the current system.

$$R_c = \left[\sum_{i=1}^{m} R(t_i)\right]^2 / k.\sum_{i=1}^{k} R(t_i)^2 \tag{13}$$

$$R_d = \left[\sum_{j=1}^{m} R(t_j)\right]^2 / m.\sum_{j=1}^{m} R(t_j)^2 \tag{14}$$

$k$ and $m$ is respectively the node number and link number of figure $R$, $R_c$ and $R_d$ can represent the fluctuation degree of the current resource saturation. When the two QoS parameters are greater, it shows that the system's average saturation degree is more balanced after accessing the service. Whereas when the bottleneck degree is lighter, the two QoS parameters are smaller, which shows that the system's average saturation degree is more imbalanced after accessing the service and the bottleneck degree is more serious. Pay attention that the domains of $R_c$ and $R_d$ are (0, 1].

2) The weighted load-balancing heuristic function.

This designs the calculation method of the load balance weighting factor based on the proportion relationship between $R_c$ and $R_d$. Assume that α is the weighted decision factor of the computing resource load balancing, β is the weighted decision factor of the transmission resource load balancing, then the calculation formulas of α and β are as follows:

$$\alpha = 1 - \frac{R_d}{R_c + R_d} = \frac{R_c}{R_c + R_d} \tag{15}$$

$$\beta = 1 - \alpha = \frac{R_c}{R_c + R_d} \tag{16}$$

Because when the values of $R_d$ and $R_c$ are bigger, the distribution of resources is more balanced, then the weight should be tend to the type of resource that with a smaller value. Thus actually formula (17) adopts the inverse proportion computation.

Carry out the linear weighting on formula (17) based on formula (15) and (16), and redefine the load balancing utility function, which is as follows:

$$q(t) = \alpha \frac{\Delta h(t)}{hu(t)} \quad \beta \frac{\Delta u(t)}{h_c(t)} \quad = \frac{R_d}{R_c + R_d} \times \frac{\Delta h(t)}{hu(t)} \quad \frac{R_c}{R_c + R_d} \times \frac{\Delta u(t)}{h_c(t)} \tag{17}$$

The Process of the F-LBQAR Algorithm

This section will describe the concrete process of F-LBQAR algorithm, which will use the $D_{ijkstra}$ algorithm to search the shortest path. The algorithm will adopt the adjacency matrix method, thus its computational complexity is $O(n_2)$, the same as the QUEST, PBSP and RBLCP. And the specific steps are as follows.

F-LBQAR algorithm / the fair load-balancing QoS routing algorithm

Input: the service request $T$, the QoS requirement of the transmission resources is $b_t$, and the QoS requirement of the computing resources is $u_t$, the overlay network $R$

Output: Find a single cast service route from node $K$ to node $M$ on $R$

1) According to the resource QoS constraint of $T$, cut off all the nodes and links of G that cannot satisfy B t and Ct, and obtain the alternative network G'. If the target nodes $K$ and M does not belong to G', then the algorithm is finished, and the business $T$ is refused

2) Start to search the shortest path based on the $D_{ijkstra}$ algorithm

3) Initialize the costs and routing information between $K$ and all the nodes, set $\forall_i \in R'$, and $DISTANCE|t| = -1$ are the distance matrix table, ROUTE is the routing list of node $K$

4) Set the root node $ROOT = K$, the search node $SN = K$, the alternative search node $B + SERARCH\{R' - k\}$, and start the iterative search for the shortest path

5) Set NB is the neighbor nodes set of SN, then there are $\forall$ node, $SN \to S$ link and $B + SERARCH\{R' - k\}$ path

6) Successively calculate the transmission resource cost weight $\Delta h(t)$ and the computing resource cost weight $\Delta u(t)$ of the node's PATH from ROOT to NB

7) Successively calculate the transmission resource fair index $d \in t$ and the computing resource fair index $d_y(t)$ of the node's PATH from ROOT to NB

8) Calculate the overall fair indexes $R'$, $R_c, R_d$ and the weighted coefficients $\alpha$ and $\beta$ of $R'$

9) Successively calculate the load-balancing weight $S \in NB$ from ROOT to all the $\Delta h(t)$ paths, if there is $\Delta h(t) \prec DISTANCE[S]$, then update the distance matrix $DISTANCE[S] = hu(t)$ and the routing list $ROUTS[S] = SN$

10) If there is $\exists N \in B\_SEARCH$, and it is the current minimum distance value of ROOT, then set the value of SN to $B\_SEARCH = B\_SEARCH - N$, and skip to step 5, otherwise to step 11

11) If there is $DISTANCE[M] = \infty$, then the search is failed, and the business T is refused

12) If there is $DISTANCE[M] \neq \infty$, then accept business $T$, and backtrack to build the service path from $K \to M$ according to $ROTUE[s]$.

## 3. Experimental Simulation and Analysis

*A. The Experimental Environment*

Based on the NS2 simulation tool, this paper simulates the resource scheduling scene of the overlay network system to verify the performance difference between LBQAR algorithm and the traditional algorithm. The simulation method adopts the NS2 platform to generate the simulative physical topology, then based on this topology builds the overlay network space through the logic application connection between the nodes, and at the end deploys the application layer routing protocol to implement the related algorithm. The specific steps are divided into two steps:

(1) The virtual mapping between the overlay space and the physical topology. Firstly, build a set of triple vectors G= {Overlay Graph, Overlay Node, Overlay Link} to describe the three kinds of entity objects of the overlay network space, which are correspondingly the overlay topology object, the overlay node object and the overlay link object, and makes these elements can be instantiated in TCL through the inheritance of the $NSO_{bject}$ object in NS2. Then establish the mapping relationship between the physical topology space in the NS2 and the overall objects, generate several Overlay Node objects by one-to-one correspondence of the actively chosen physical nodes, based on the underlying routing table and using the adjacent lead connection [12], which is to make there are no other overlay nodes in the overlay link between the overlay nodes, to construct the

overlay network topology and generate the corresponding Overlay Link objects. At last, form the overlay topology objects Overlay Graph through the combination of the Overlay Node and the Overlay Link.

(2) The real-time Agent monitoring. After the generation of the overlay network, it is still necessary to adopt the end-to-end measurement method to obtain the all kinds of network performance parameters in the overlay network. First of all, inherit the Agent object in NS2 to generate the Overlay Agent object, which can be placed at any Overlay2Node and evaluate and exchange the performance of the Overlay Link and the Overlay Node through sending the explorer packet to the neighboring nodes, such as the transmission delay, the path available bandwidth and so on. All of the real-time performance data and communicated connection information can be stored in the performance database of the Overlay Graph, and according to these information, the Overlay Graph object can real-time calculate the overall routing list.

For the business communication improves the flow mechanism of NS2, this paper designs the Overlay Traffic object that can establish multiple flow Applications, which is the Application object in NS2, between the Overlay Node nodes to form the connected service path in the application layer according to the routing list in Overlay Graph, and based on the established routing information stored in the Overlay Path object in the service path, the operations of the path can be carried out at any time, such as the demolition, interrupt, update and so on.

*B. The Analysis of the Experimental Results*

This experiment is divided into two groups. One is to measure the effects of the algorithm on the system's throughput under the increasing of the network size and traffic. The other is to measure the effects of the algorithm on the system's load fluctuation degree under the increasing of the network size and traffic.

Test on the Service Success Rate and Throughput

Experiment 2 simulates the scene that massive QoS service requests continuously arrive in, and through the service success rate, throughput and other QoS indexes, it validates the effects of different load balancing overlay routing algorithms on the system throughput. The experimental environment is based on GT-ITM to respectively generate 4 kinds of randomly distributed physical networks, whose sizes respectively are 50 nodes (scale = 100 x 100), 100 nodes (scale = 100 x 100), 150 nodes (scale = 100 x 100), and 200 nodes (scale = 100 x 100). The probability that there is a two-way link between the nodes is 0.02, the bandwidth of each link is 2Mbps, and the delay is 200~1000 m s. Choose the 30% of the physical nodes as the overlay nodes, and the largest computing capacity of each node is 20unit/s. Simulate the CBR traffic flow with a arrival Rate of 20 ~ 408 unit/s through Overlay Traffic, and the average flow is 300kbps. Respectively test the QoS service success rate (ASSR) of the QUEST, PBSP, RBLCP and F-LBQAR algorithm.

$$QSSR = \frac{\text{The number of sessions of the accepted requests}}{\text{The number of sessions of the arrived services}} \quad (18)$$

The experiment adopts the sparse degree of physical network with the probability of 0.04 to construct the overlay network, thus it is relatively easy for the limited network resources to cause the congestion. It can be observed obviously from the data in figure 4 the process that the system gradually becomes overloaded and the success rate gradually decreases. The analysis of the experimental results shows that with the constantly growth of the business flow, the QoS service success rate of the resource scheduling constantly falls, which suggests that there are congestion and overload in the network. Due to the load-bearing ability of the network usually improves with the increasing of the network scale, thus the network overload points of the 50 nodes in figure 4 (a) appears relatively earlier, which is around at 80 unit/s load, while the overload points in figure 4 (d) appears relatively late, which is about at 260 unit/s load. It can be seen from the experimental results that no matter under what kind of topology, compared with the traditional algorithm, the F - LBQAR algorithm can increase the QSSR by 1% ~ 6% or so under the overloading of the system, it is especially more apparent for small networks, and it can improve the success rate by about 6%, which is show in figure 4 (a) and at 121 ~ 205 unit/s. The results show that compared with the traditional algorithm, the F - LBQAR algorithm is more adaptive to bad network environments and it is conducive to improve the throughput of the network system.

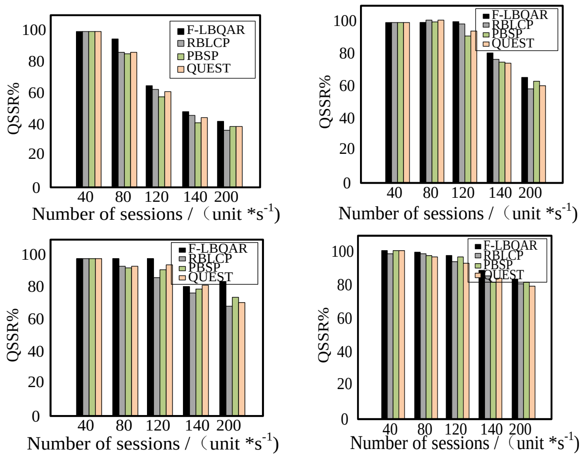
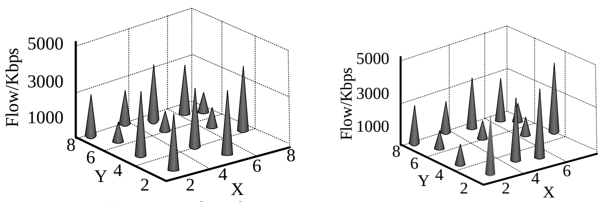



**Figure 4. The Experimental Results of the QSSR Test in Different Topologies**

The experimental results of figure 5 describe the flow distribution of the network nodes which are formed when implementing the various algorithms and the load of the system reaching at 100unit/s business volume in the network plane with a size of 100 nodes. It can be seen that each algorithm still produces the nodes with congestion, and the flow values of the nodes are relatively higher. But after the execution of the F-LBQAR algorithm, it has relatively less congestion nodes than other algorithms, and the load can be more evenly distributed in the network plane. Table 1 counts the distribution of the nodes number in the flow peak under the situation. It can be known by analysis that under the experimental environment, the probabilities of high-loaded nodes in the flow interval of [5009 kbps, 4005kpbs], [4007 kbps, 3008 kbps] and [3000 kbps, 2000kbps] of the QUEST algorithm and PBSP algorithm are much higher than that of F-LBQAR, and the peak nodes of the F-LBQAR algorithm are mainly concentrated in the interval of [1002kbps, 0kbps], which shows that the F-LBQAR algorithm can more balanced distribute the load, namely improves the utilization degree of the network resources, and reduces the number of the congestion nodes at the same time.




(a) QUEST algorithm (b) PBSP algorithm

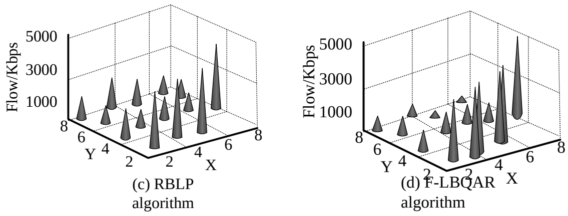



**Figure 5. The Distribution of the Node's Throughput of Each Algorithm with the Network Size=100 and 100 unit/s**

**Table 1. The Statistics of the Nodes in the Flow Peak**

| The range of the throughput | The number of the nodes | | | |
|---|---|---|---|---|
| | PBSP | RBLCP | QUEST | F-LBQAR |
| [5009kbps, 4005kpbs] | 4 | 4 | 4 | 0 |
| [4007 kbps, 3008 kbps | 4 | 1 | 4 | 5 |
| [3000 kbps, 2000kbps] | 16 | 13 | 18 | 5 |
| [2002kbps, 1000kbps] | 36 | 38 | 43 | 44 |
| [1002kbps, 0kbps] | 48 | 47 | 49 | 53 |
| Total | 100 | 100 | 100 | 100 |

The Fluctuation Degree Test on the Load-balancing

Experiment 3 simulates the scene when the massive QoS service requests continuously arriving in, and tests the resource saturation volatility situation of the links and nodes in the system to verify the load balancing ability of the different overlay routing algorithms. Through the two QoS indicators: the Residual Link Capacity Deviation (RLCD) and the Residual Node Capacity Deviation (RNCD), the experiment measures the influence of the algorithm on the system's overall load balancing degree. And the definition is as follows:

$$RNCD = \sqrt{\sum_{i=0}^{k}\left(h\left(t_i\right)-u_4^2\right)/k} \quad (19)$$

$$RLCD = \sqrt{\sum_{i=0}^{k}\left(u\left(t_i\right)-u_4^2\right)/k} \quad (20)$$

$h\left(t_i\right)$ and $u\left(t_i\right)$ are the resource saturation functions of the physical nodes and links in the network, c/4 and b/4 are the average saturations of the two kinds of resources in the system, k and m are respectively the total number of the nodes and links.

Firstly, based on GT-ITM, the experiment respectively generates two kinds of randomly distributed physical networks, and the sizes of the networks are both 100 nodes. The probability that there is a two-way link between the nodes in one network is 0.1, the scale is 10×10, and the bandwidth of each link is 2Mbps. Randomly set the 32% of the physical nodes are the overlay nodes, and the largest computing capacity of each node is 20unit/s, then the network belongs to the network topology with a relatively sparse computing resources. The probability that there is a two-way link between the nodes in the other network is 0.2, the scale is 100×100, and the bandwidth of each link is 1Mbps. Randomly set the 34% of the physical nodes are the overlay nodes, and the largest computing capacity of each node is 40unit/s, then the network belongs to the network topology with a relatively sparse transmission resources. Simulate the CER business flow with an arrival rate of 0~256unit/s through the Overlay Traffic, the average flow is 300kbps, and respectively test the RNCD and RLCD of the QUEST, PBSP, RBLCP and F-LBQAR algorithm under the condition of different topologies and loads. The experimental results are shown in figure 6 and figure 7.

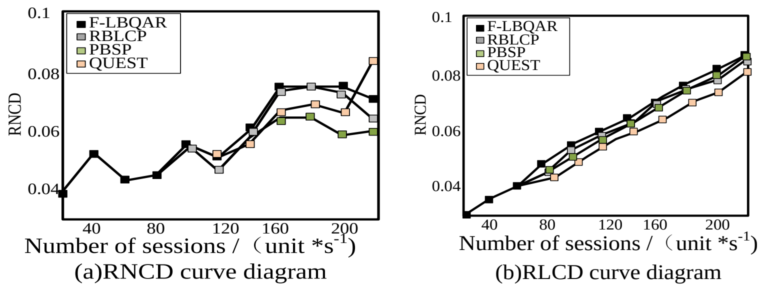



**Figure 6. The Test of Load-balancing Volatility in the Computing Resource Bottleneck with**

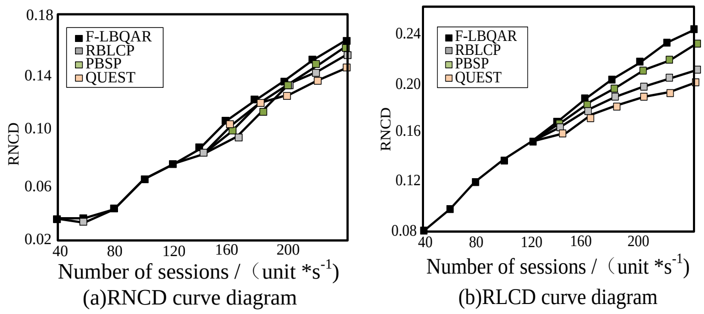



**Figure 7. The Test of Load-balancing Volatility in the Transmission Resource Bottleneck with 100 Nodes**

It can be observed from the experimental results of figure 5 and figure 6 that with the increasing of the load, the volatility of all the resource saturations will also be on the rise, which shows that the load will increase the imbalance degree of the system. While after the occurrence of the congestion overload, the volatility will fall, this is because the overloaded service nodes and links will reject the new request, resulting in the load being distributed to the free resources naturally. Figure 5 describes the equilibrium conditions of the computing resources relative to all kinds of algorithms on a relatively sparse network, it can be seen that there appears obvious congestion, which is more than 140unit/s, in the computing resources of the network in the case of the high loading, while the volatility of the transmission resources is still on the rise, which shows that the transmission resources are still not saturated. But the results of figure 7 show that in the network environment with relatively scarce computing resources, relative to the traditional algorithm, the F-LBQAR algorithm can reduce RNCD by about 3% ~ 8% under the condition of the system's overloading, and its RLCD value basically keeps in the same level with other algorithms. The results of figure 7 are similar to the results of figure 5, but due to the congestion nodes will affect multiple links, the transmission congestion of figure 7(b) is less obvious than that of figure 5(a). However, the results still show that the F-LBQAR algorithm can still reduce RLCD by about 2%~6% under the condition of the system's overloading, the reducing is more than 160 unit/s, and its RNCD value basically keeps in the same level with other algorithms.

It can be concluded by analyzing the results of the two kinds of experimental data that relative to the traditional algorithm, the F-LBQAR algorithm has a better ability to evenly distribute the resources, and it is especially beneficial for the optimization of the distribution situation of the system's scarce resources. And under the high-loaded business environment and with the same throughput flow, it can reduce the volatility of the resource saturation (RLCD, RNCD) by about 2% ~ 8%.

## 4. Conclusion

To solve the load-balancing QoS routing of the distributed application under the overlay network environment is a very important research direction. The traditional overlay network QoS routing algorithm fails to consider the effects of the resource bottleneck on the routing selection, which is easy to cause the local congestion in the system. Thus this paper proposes a fair load-balancing QoS overlay routing algorithm F-LBQAR, which can reduce the probability of the network congestion caused by the resource bottleneck, and improve the QoS service success rate and the throughput of the system.

## References

[1] M. J. Mirza and N. Anjum, "Association of Moving Objects Across Visual Sensor Networks", Journal of Multimedia, vol.7, no.1, **(2012)**, pp. 2-8.
[2] D. Andrea, B. Melazzi and S. Stefano, "CONET: A content centric inter-networking architecture", In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking (ICN'l 1), **(2011)**; Canada.
[3] M. Tao and Z. Chunhong., "On the disruptive potentials in internet of things", In Proceedings of 17th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2011), **(2011)**; Taiwan.
[4] J. Van, D. K. Smetters, J. D. Thornton, "Networking named content", In Proceedings of the 2009 ACM Conference on Emerging Networking Experiments and Technologies(CONEXT'09), **(2009)**; Italy.
[5] S. Stefan, C. Andy and H. Eric, "End-to-end effects of Internet path selection", ACM SIGCOMM computer communication review, vol.29, no.4, **(1999)**, pp.289-299.
[6] M. Tao, Z. Chunhong and Z. Zhimin, "Selecting relays for faster paths in embedded internet delay space", International Journal of Advancements in Computing Technology, vol.4, no.2, **(2012)**, pp.176-184.
[7] D. Leonard and D. Loguinov, "Turbo king: Framework for large-scale internet delay measurements", In Proceedings of the IEEE INFOCOM 2008, **(2008)**; USA.
[8] M. Tao and Z. Chunhong, "A power law relay routing overlay", In Proceedings of 4th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP 2011), **(2011)**; China.
[9] H. Zheng, E. K. Lua and M. Pias, "Internet Routing Policies and Round-Trip-Times", In Proceedings of Passive and Active Network Measurement, **(2005)**; USA.
[10] G. Wang and B. Zhang, "Towards network triangle inequality violation aware distributed systems", In Proceedings of 7th ACM SIGCOMM Internet Measurement Conference, **(2007)**; USA.
[11] Y. Geng, J. Chen and K. Pahlavan, "Motion detection using RF signals for the first responder in emergency operations: A PHASER project", 2013 IEEE 24nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), **(2013)**; London, Britain.
[12] S. Li, Y. Geng, J. He and K. Pahlavan, "Analysis of Three-dimensional Maximum Likelihood Algorithm for Capsule Endoscopy Localization, 2012 5th International Conference on Biomedical Engineering and Informatics (BMEI), **(2012)**; Chongqing, China.
[13] M. Tao and Z. Chunhong, "Topology proximity and property-aware distributed hash tables based on network coordinates", Advances in Information Sciences and Service Sciences, vol.4, no.7, **(2012)**, pp.308-316.
[14] S. Zhou, G. R. Ganger and P. Steenkiste, "Location-based Node IDs: Enabling explicit locality in DHTs", Carnegie Mellon University School of Computer Science Technical Report CMU-CS-03-171, **(2003)**.
[15] G. Yang, S. Kyoungwon and K. Jim, "P2Cast: Peer-to-peer patching for video on demand service", Multimedia Tools and Applications, vol.33, no.2, **(2007)**, pp.109-129.
[16] J. Buford, A. Wang and X. Hei, "Discovery of In-Band Streaming Services in Peer-to-Peer Overlays", In Proceedings of GLOBECOM'07, **(2007)**; USA.
[17] K. Sungsu, S. John and H. J. Won-Ki, "Semantic overlay network for peer-to-peer hybrid information search and retrieval", In Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management, **(2011)**.

# Authors

**Zhou Xiao-Yan**, she was born in HeBei Province, China, in 1979. She got the master degree of engineering from the Ocean University of China, in 2006 .Her research interests include: computer network security and Artificial intelligence. She is currently a lecture of Information Engineering Department of Karamay Vocational and Technical College, Karamay, Xinjiang, China.