International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 6 Issue 5|| May 2017 || PP. 26-34

An Approach using Local Information to Build QoS Routing Algorithm

Nguyen Chien Trinh[1], Tran Minh Anh[2]
[1,2]*Posts and Telecommunications Institute of Technology, Hanoi City, Vietnam*

***Abstract*:** *The requirement for quality of service (QoS) is more and more sophisticated, such as the required bandwidth, the value of delay time or packet loss. To assure the network performance, localized QoS routing algorithms have recently been proposed as a promising alternative to the currently deployed global QoS routing schemes. Different from the traditional QoS routing algorithms which use global state information, the localized routing algorithms use local information collected from source node to make routing decisions. These localized routing algorithms can be solutions to users' demand in the near future. In this paper, we propose a new localized QoS routing algorithm which can help to assure quality of service, and show our simulations which are better in results against other routing algorithms.*
Keywords: *Algorithm, delay, localized, packet loss, routing.*

I. INTRODUCTION

In the near future, telecommunication (telecom) services like multimedia, online services … require very high speed and high quality transmission. To satisfy this demand, the technique of using local information at source node to build sets of paths available for routing is an effective method to communicate in network. This routing technique will decrease the average overall blocking probability of packets transmitted through the network as well as improve the overall performance of network. It has been performed that this technique is simpler and better than traditional global QoS routing schemes like widest shortest path (WSP). Because of basing on local information, it routes flows in network with localized view of the network QoS state, so it avoids some problems committed by traditional global QoS routing schemes which update periodically by a link-state algorithm and maintaining it up-to-date, such as: a large communication overhead, the inexact of global state and the out-of-date information due to large update intervals.

Localized QoS routing method otherwise tries to avoid these problems by routing based on local information, it means it performs flow routing using this localized view of the network QoS state. In this approach, each node builds and maintains a predetermined set of candidate paths to each possible destination and routes flows along these paths. There are many ways of selection of the candidate paths which impact on how the localized routing algorithm performs and differ from one another. Anyway, the localized method will open a new approach to assure QoS for network in the near future.

In this paper, we propose a novel routing algorithm which uses three parameters of QoS (bandwidth, delay and bit-error-rate) as criteria for routing, with an index of path to choose best path from set of candidate paths. We compare and realize better performance against other localized routing algorithms and the traditional global QoS routing algorithm Widest Shortest Path (WSP) when we perform simulations with the same types of topology, traffic patterns and under the same range of traffic loads.

The rest of the paper is structured as follows. Section II introduces some QoS routing approaches and CBR, one of localized routing scheme. Section III describes our novel routing algorithm with its flowchart and pseudo-code. Section IV studies the performances of simulations which show the better results of our routing algorithm than other algorithms. And in the end, section V concludes the paper and proposes the next work.

II. RELATED WORKS

Recently, parameters of QoS have been used to evaluate the quality of transmission for telecom services in some years recently. Up to now, there are a lot of researches for QoS routing which have been published on many different areas as [1-5]. In the future, when the convergence of services becomes stronger, QoS routing will be more important. One of the QoS routing algorithms is widest-shortest-path (WSP) if proposed in [2] by Guerin et al.. In WSP, a modified Bellman-Ford algorithm is used to pre-compute paths from a node to all destinations in the network. For each destination, WSP computes paths of all possible bandwidth values, and uses them to build a QoS routing table which is kept separate from the standard OSPF (Open Shortest Path First) routing table. This routing table generated by the algorithm can be conceptually viewed as a matrix, where a row corresponds to a destination (entry in the IP routing table), and the i^{th} column corresponds to paths that are no longer than i hops and have the largest amount of bandwidth available among all such paths to the specific destination. The information stored in a matrix entry includes the next hop(s) and the available bandwidth on such paths.

The information in the QoS routing table is used to identify paths capable of satisfying the bandwidth requirements of new requests. This is accomplished by comparing the amount of bandwidth requested by a new flow to the available bandwidth in successive entries in the row associated with the flow's destination. The search stops at the first entry with an available bandwidth value larger than the requested one, at which point the corresponding next hop is returned and used to determine the next hop on which to forward the request. If there is more than one next hop, our implementation chooses one of them at random with a probability that is weighted by the available bandwidth on the associated local interface corresponding to the next hop. By this way, WSP becomes very popular in the network recently.

Next, as mentioned in the former chapter, localized QoS routing is a new approach which bases on the idea of using local information to route packets in the network. One of these schemes is the scheme of the localized Credit Based Routing algorithm (CBR) proposed by [6]. As well as other localized routing algorithms, the CBR predetermined a set of candidate paths R between each pair of source and destination from *minhop* set R_{min} and alternative paths set R_{alt}. In the algorithm, the CBR build a crediting scheme to reward and penalize a path based on the statistics about flow routing in the network.

The CBR updates frequently the credit for all paths after its times of transmitting flow. If a path transmits successfully a flow, its credit will increases and vice versa. The value of credit finally reflects the flow blocking probability of that path, and this mere value is used to select path itself for next incoming flow. The larger the credit of path is, the more chances for that path to be selected are. It means that the CBR selects the path with largest credit in each set R_{min} and R_{alt} upon flow arrival. Then, the CBR uses the system parameter Φ ($\Phi \leq 1$) to decide the route for flow with the formula as below:

If $R_{min}.credits \geq \Phi \times R_{alt}.credits$, the flow is routed along the minimum hop path R_{min}. If $R_{min}.credits < \Phi \times R_{alt}.credits$, the flow is routed along an alternative path R_{alt}.

Based on the statistics of the path blocking probability, the CBR increases or decreases the path credits upon flow acceptance or rejection. Besides, with that blocking probability, the CBR fixes a MAX_CREDITS parameter to determine the maximum attainable credits for each path, as follows:

$$0 \leq R.credits \leq \text{MAX\_CREDITS} \quad (1)$$

To do that, the CBR algorithm uses a sliding window for M connection requests, and records rejection and acceptance for each path. It uses 1 for flow acceptance and 0 for flow rejection in that window. With the value taken from this window, the CBR will calculate each path blocking probability for the period of M connection requests. The main problem with CBR is that a path's credits are only updated each time that path is selected. If a path is selected infrequently, its credit value will become stale leading to errors in the selection process. The schemes above will be used to compare with our proposed algorithm through simulations. The results of simulations will be showed in the section IV.

III. A PROPOSED LOCALIZED QOS ROUTING ALGORITHM

3.1. Methodology

QoS routing is gradually becoming an essential part of today's data networks as many applications depend on QoS routing to provide the promised quality services. QoS routing uses network state information and resource availability in addition to the QoS requirements to meet such demands. QoS routing algorithms that select paths with sufficient residual resources to meet the QoS constraints have played a critical role in meeting the required service level. Therefore, network providers must assure the quality of network for customers who use these high quality services. For that reason, in the recommendation Y.154x as in [7], ITU-T has proposed the requirements for QoS standard as Table 1.

Table I. Class of services and QoS standard

Class Application/Examples	Mean Delay upper bound	Delay variance	Loss Ratio
Class 0 Real-time, jitter-sensitive, high interaction (VoIP, Video Teleconference)	100ms	<50ms	<1.00E-3
Class 1 Real-time, jitter-sensitive, interactive (VoIP, Video Teleconference)	400ms	<50ms	<1.00E-3
Class 2 Highly interactive, transaction data (e.g signaling)	100ms	Unspecified	<1.00E-3
Class 3 Interactive transaction data	400ms	Unspecified	<1.00E-3
Class 4 Low loss only (Short transactions, bulk data, video streaming)	1s	Unspecified	<1.00E-3

Class 5 Default IP networks applications	Unspecified	Unspecified	Unspecified

In the table, we can see that IP networks are now carrying a heterogeneous mix of traffic, with widely differing QoS requirements. The service model of emerging multi-service packet networks is based on the network's ability to guarantee QoS to user applications. Besides bandwidth and delay, end-to-end packet loss is one of the most significant QoS performance metrics for customer's services because for many applications, the performance drops drastically if the end-to-end loss exceeds a certain limit, and will become unusable under excessive loss. Therefore, in this paper, we take these three metrics Bandwidth, Delay and bitErrorRate as criteria in our scheme, then we call our scheme as Bandwidth-Delay-bitErrorRate based Routing or BDER.

3.2. Describing the algorithm of BDER:

Like other localized algorithms, our algorithm BDER also requires every node to maintain a predetermined set of candidate paths R to each possible destination, but unlike CBR which distinguish between two types of paths *minhop* paths and *minhop+1* paths, BDER only builds and keeps track for only one set of candidate paths R (the result of finding shortest paths). In our scheme, we propose every path $P_i \in R$ associated with a variable P_i.Quality [1, 2, 3] =P_i. [Bandwidth, Delay, BER] and an index of that path, called T_i: total flow accepted on that path between source and destination. Call T as sum of flows requested from source node to a specified destination. Then the ratio T_i/T will be used for ranging set R, it means range R $\{T_i/T\}$. And, to choose path, we choose the path in set R with maximum value of (T_i/T). If there are a lot of paths having the same value of (T_i/T), we choose the next criterion: *min(*BER*).* All the procedures of choosing path like that, we call shortly *max*(R).

Routing process: Upon flow arrival, BDER does some steps:

1. Selects the path P_i with *max*(R) (first place in set R after ranging)
2. Check the demand of the flow and use it for comparing with the value of path, we call that is RQ.[Bandwidth,Delay,BER], or RQ (Requested Quality).
3. Compare P_i. Quality and RQ (the demand)

- If P_i. Quality ≥ RQ, the P_i will be chosen. The way to compare will be discussed in part of 3.4.
- Else: select the next P_i in R (the *max*(R) of the rest of R)

4. The loop will be done until finding out the path has maximum of R and P_i.Quality ≥RQ.

- If no path satisfies RQ, the arriving flow obviously is cancelled.

5. Increase T_i and T if path P_i is accepted, keep T_i, increase T if path P_i is unaccepted accordingly.

- If no path is chosen, we increase T only.

Unlike CBR, BDER changes the ratio $\{T_i/T\}$ of this path. The index T_i, the index of path P_i, increases 1 when it is accepted for a flow-in. Then, the ratio T_i/T will be used to compare among paths for the next flow-in. One path has highest ratio, it means it has most "potential" in the set R, and it will be chosen for the next flow-in. After that, the following flow will use this ratio $\{T_i/T\}$ to be criterion to choose path, and next loop re-begins. Therefore, after one flow is processed, the ratio $\{T_i/T\}$ changes accordingly to the success/fail rate of that path, and the "value" of that path changed correspondently. It affects to the probability of being chosen for the next.

3.3. Flow chart of all steps:

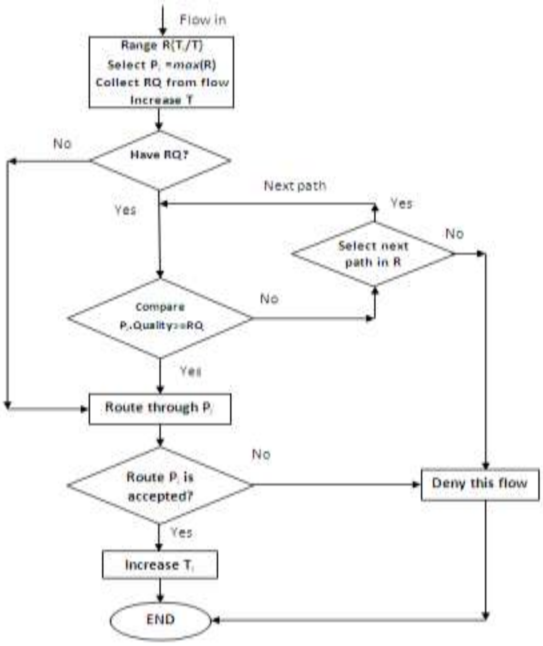

Figure 1. Flow chart of BDER

3.4. The metric selection and comparison

As mentioned before, we choose Bandwidth, Delay, and bitErrorRate (BER) as the three metrics for comparison to choose paths. There are some ways to compare, but the relevant and popular way will be discussed as follows:

We make the comparison of three metrics independently. First, we compare the bandwidth of the path with the demanded bandwidth of flow-in. If it's true (i.e. the value of network can afford the value of flow-in), we continue to compare the Delay of that path with the demanded delay for that flow. At the end, we compare the value of bitErrorRate (BER) of the path with the value of the flow request. If we have three values of true from all three comparisons, the path is proved "enough" quality for flow, and the path will be chosen. (Note : We compare as shown in Fig 2: P_i.Bandwidth≥RQ.Bandwidth; P_i.Delay≤RQ.Delay; and P_i.BER<=RQ.BER; where Bandwidth is determined: the minimum residual bandwidth of any link on path, Delay of path: the sum of all propagation delay from all links on this path and BER of path is sum of all BER on all links of path).

```
PROCEDURE Compare3 (p,f)
If p.Bandwidth ≥ f.Bandwidth
    If p.Delay ≤ f.Delay
        If p.BER ≤ f.BER
            p is chosen
        Else
            p is discarded
    Else
        p is discarded
Else
    p is discarded
END PROCEDURE
```

Figure 2. Using comparison with three constraints

3.5. The pseudo code of algorithm:

```
Initialize
Building R
Set T=1
Set Ti=0, ∀ P∈R
BDER
1. Range R on {Ti/T} for flow-in
2. Set P.success = false;
3. Set RQ for flow-in
4. P=first(R)
5. Increase T
6. Do while !(P.success or R(end))
7. if(P.Quality>=RQ)
8. Route flow along path P
9.   if P is accepted
10.   {Increase Ti
11.   P.success=true}
12.  elseif
13.   Flow is cancelled
14.  endif}
15. ElseIf
16. P= next(R)
17. Endif
18. Loop
19. END.
```

Figure 3. The BDER pseudo code

IV. PERFORMANCE EVALUATION

In this section, we realize the performance of the BDER scheme and compare it with other schemes like CBR and the global QoS routing scheme widest shortest path (WSP) which searches for a feasible path with minimum hop count. All the experiments will be set in the same condition.

4.1. Simulation Model

We use OMNeT++ [8], a common simulator now. To evaluate the results, we collect all of parameters of simulation as vectors, scalars and histograms to compare. The setup of simulation experiments is described as follows: network built after the network of an famous ISP in USA with 60 nodes, links of these nodes are all bidirectional with the same capacity C = 150Mbps in each direction and the same value of delay D = 10ms, and bit error rate (BER) of 200E-6, flows arrive to each source node according to a Poisson process with rate λ and destination nodes are selected randomly (each node can be source or destination), flow duration is exponentially distributed with mean $1/\mu$, bandwidth of flows is uniformly distributed within [0.5-4MB].

As analyzed in [9-10], the offered network load is $\rho = \lambda Nbh/\mu LC$, where N is the number of nodes, b is the average bandwidth required by a flow, h is the average path length (in number of hops) and L is the number of links in the network. In the experiments, we have N=60, L=158, h=5.812, $1/\mu$= 60s. Since the performance of routing algorithms may vary across different load conditions, our simulation experiments consider several types of different load conditions through the value of λ according to experiments of from low loads to high loads.

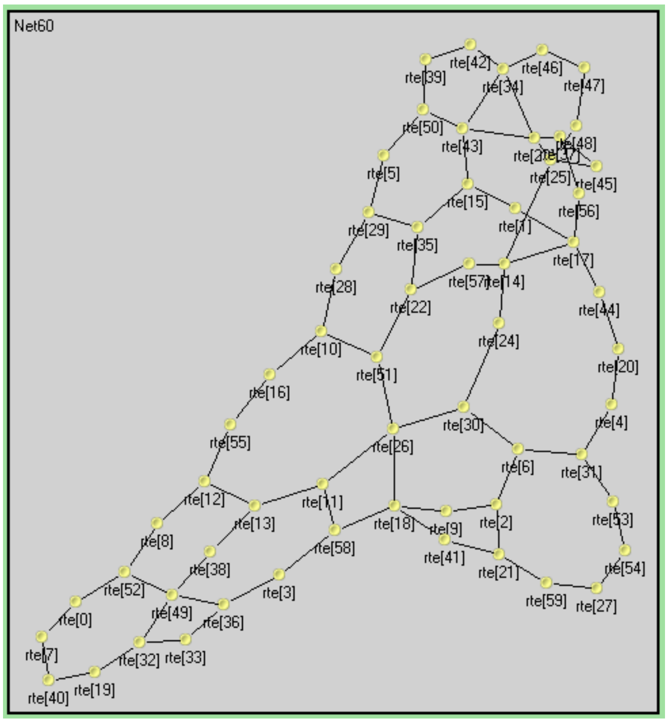

Figure 4. Network of 60 nodes

To compare with other schemes, we choose flow blocking and bandwidth blocking probabilities as criteria as well as the simulation in [6, 9-10]. The blocking probabilities are calculated based on the most recent 100,000 flow after more than 2.5 million of flows emitted. We take 2 ratios as:

$$\text{Flow Blocking Probability} = |B|/|T| \qquad (2)$$

where |T| is the total of all flows, |B| is total of blocked flows.

We also calculate the overall end-to-end delay of network when we run the scheme BDER and the scheme of WSP with different load. Otherwise, we calculate and survey the impact of bursty traffic on the operation of algorithms. From those results, we can conclude the effectiveness of the BDER scheme against other schemes.

4.2. Simulation Results

4.2.1. Flow Blocking Probability and Bandwidth Blocking Probability:

With the results of blocking probability collected from the simulation, we compare with the ones of other schemes, as shown in Fig.5.

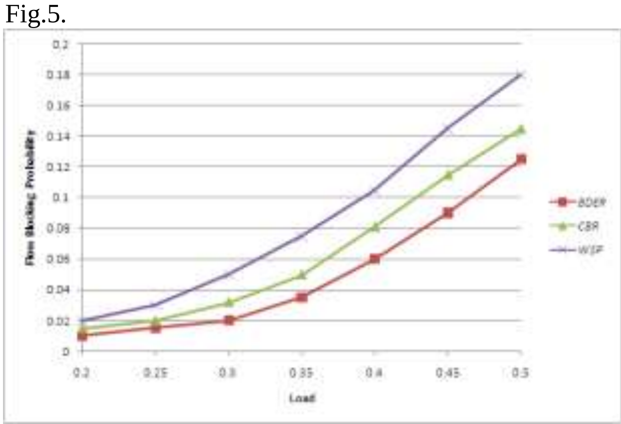

Figure 5. Flow Blocking Probability

Fig.5 shows the performance of BDER against CBR and WSP in terms of flow and bandwidth blocking probabilities under load ρ varies between 0.2 to 0.5. From these values, we see that under low load ($\rho \leq 0.25$), the difference in the performance of the routing algorithms is quite small, because finding available path with sufficient bandwidth is easy and flows are almost accepted. When ρ is high (more than 0.3), some differences reveal. Many flows drop and fail to get destination node, then flow blocking probability grows rapidly, as viewed in Fig.5.

To see why this is, in the case of the BDER scheme, paths are selected based on the value $\{T_i/T\}$ of predetermined paths, then this value increases assures that this path is good and may support relatively the next flow. Other than CBR, BDER uses Bandwidth, Delay and BER as criteria to compare, hence all flows which are routed along the chosen path, almost satisfy the demand of Bandwidth, Delay and BER at destination, so it helps diminishing flow blocking probability of that path, and also diminishing the value of End-to-End Delay of those flows. Moreover, the index of path is set to avoid the congestion of flows which come at nodes simultaneously, particularly when the load increases and the links become congested. If congestion happens, flows will be re-directed to other path and the index decreases at once. Then, the source node might diminish using of these paths which have low index. Therefore, the probability of flow blocking is considerably low against the case of CBR and WSP as well. Next, we collect information of Average End-to-End Delay with load $\rho=0.8$ as in the Fig.6,

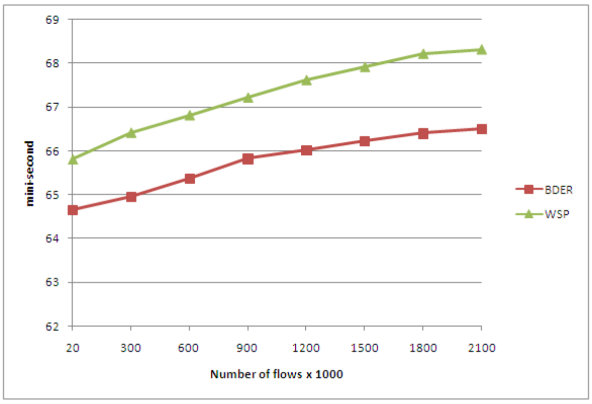

Figure 6. Average End_to_End Delay of Flows when $\rho=0.8$

From the Fig. 6, we can see that when the number of flows increases, the Average End-to-End Delay will keep a stable value and the BDER expresses the more efficiently. It means that with high load, the congestion happens more frequently, so, this value is higher. In our case, the BDER changes path more frequently based on the ratio $\{T_i/T\}$. When congestion happens, the index of the candidate path diminishes, then, our case changes path. Therefore, the average End-to-End Delay is better than case of WSP as viewed in Fig. 6.

4.2.2. Impact of bursty traffic on the operation of algorithms

In the Internet, telecom services and applications use bursty traffic quite much. After some researches such as [11-12], we can see that the routing performance with blocking probability go down when the traffic burstiness increases, and the burstiness of traffic is shaped by the distribution of Weibull. In this paper, we also use Weibull distribution with the shape value of 0.4 to expose the bursty flow arrivals.

Fig. 7 shows the flow blocking probability plotted under a range of offered loads from 0.2 to 0.5 with the shape value of 0.4 for all algorithms. At low loads, all algorithms perform almost the same. This is because links are lightly used; which would allow for further bursty flows to be accepted. At high loads, the algorithms perform differently. With CBR, CBR makes routing decisions based on blocking probability, so the value of blocking probability increases with burstiness. This relatively slow response to changes in bursty traffic causes more blocking, and hence, higher blocking probability. The WSP is worse than BDER, because it infers frequently the global state information to compute the route for flow, then with bursty traffic, it commits blocking increasingly by making harder to find feasible paths. With BDER, it uses set of paths to transfer flow through network, so it has the most favorable values as showed in the Fig. 7.

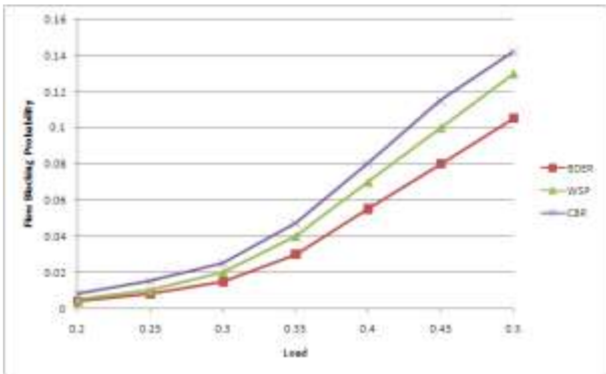

Figure 7. Impact of bursty traffic.

In concluding, the case of BDER has better performance than other cases such as CBR or WSP in some experiments which have been done.

4.3. Complexity and overhead

The case of WSP uses the algorithm Dijkstra, like almost global QoS routing algorithms, takes at least O(NlogN+E) time, where N is the size of the network measured in the number of nodes, and E is the number of links (edges) as in [3]. At the same time, the localized schemes use the way of routing that selects path from the set of candidate paths R, with the size of R that is |R|.

In the CBR algorithm, the path selection is an invocation of a weighted-round-robin like path selector (wrrps), whose worst case time complexity is O(|R|) as [6], similarly bder requires order of better than that, for it uses only the minhop of paths as explained above. In addition these localized schemes require updating information, which takes a constant time O(1). Therefore, with communication overhead, bder or other localized schemes require very little over and above computing the blocking probability based on acceptance or rejection of a path, while at the same time, global algorithms require a huge amount of overhead to keep the link state information updated. In conclusion, the computation of localized of our case at source node anyway is much smaller than the one of traditional WSP cases.

V. CONCLUSION

In this paper, we propose a new localized QoS routing model to choose path using only flow information collected locally at source node. Many experiments have been done to compare between its performance against the CBR and the WSP algorithm; and have showed a comparable performance with better blocking probabilities, better time complexity and lower communication overhead which confirms the localized. The flow chart and pseudo code of the algorithm have been proposed with the three metrics bandwidth, delay and bit-error-rate as QoS metrics to route flows through network. Using these tools, the local information of blocking probability could be used to update the path quality through our proposed index. This index directly decides the routing, hence makes better quality of routing, on the other hand, better working of network.

As part of future work, we will investigate the ability of this algorithm in using more QoS parameters to compare, loss packet or delay jitter and so on. It will of course make a lot of computation, and be very complicated, but in the very high speed network, it will become very important, when all services will use the same infrastructure, and the telecom services will surely require very large bandwidth. And finally, we will investigate the effect of the topology of network on the operation of the localized routing algorithm. The algorithms work more effectively when the network has the more balancing topology. With the proposition of localized routing algorithm on that better topology network, the load will be distributed more equally to all links in network, and at the same time, it brings better results for network routing.

REFERENCES

[1] C. Pornavalai, G. Chakraborty, N. Shiratori, QoS based routing algorithm in integrated services packet networks, Proceedings of the IEEE ICNP, 1997.
[2] R. Guerin, et al, QoS Routing Mechanisms and OSPF Extensions, Proceedings of GlobeCOM, 1997.
[3] G.Apostopolopoulos, D. Williams, S. Kamat, R.Guerin, A. Orda and T. Przygienda, QoS Routing Mechanism and OSPF Extensions, RFC2676, 1999.
[4] Z. Whang and J. Crowcroft, Quality-of-Service routing for supporting multimedia applications, IEEE J. Select. Areas Commun, 14 (1996), pp. 1228-1234.
[5] R. Guerin and A. Orda, QoS based Routing in Networks with Inaccurate Information: Theory and Algorithms, IEEE Transaction on Networking, 7(1999), pp.605-614.
[6] S. Alabbad and M. E. Woodward, Localized Credit Based QoS Routing: Performance Evaluation Using Simulation, the 40th Annual Simulation Symposium, Huntsville, AL 2006.

[7] ITU-T Recommendations, Y.1500-Y.1599: Quality of service and network performance. http://www.itu.int/ITU-T/recommendations/index .aspx?ser=Y.
[8] A. Varga, The OMNeT++ Discrete Event Simulation System, the European Simulation Multiconference, Prague, Czech Republic, 2001.
[9] A. Shaikh, J. Rexford, K. Shin, Load-Sensitive Routing of Long-Lived IP Flows, ACM SIGCOMM 1999.
[10] A.Shaikh, J.Rexford, K.G.Shin, Efficient Precomputation of Quality-of-Service Routes. Proc. IEEE NOSSDAV 98, July, 1998.
[11] A. Shaikh, J. Rexford, and K. Shin, Evaluating the Impact of Stale Link State on Quality-of-Service Routing, IEEE/ACM Transactions on Networking, vol. 9, pp. 162-176, 2001.
[12] A. Feldmann, Impact of non-Poisson arrival sequences for call admission algorithms with and without delay, GLOBECOM'96. IEEE Global Telecommunications Conference, London, UK, pp. pp. 617-622, 1996.