Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.692 – 706

RESEARCH ARTICLE

Characteristic Evaluation of Distributed QoS Routing

Shuchita Upadhyaya[1], Gaytri Devi[2]
[1]Department of Computer Science and Applications, Kurukshetra University, Kurukshetra
[2]GVM Institute of Technology and Management, DCRUST, Sonipat
[1] shuchita_bhasin@yahoo.com, [2] gayatri.dhingra1@gmail.com

ABSTRACT
In current scenario of Internet, the demand of real time multimedia applications has been increased. These applications are bandwidth greedy, impose strict delay guarantees, stable jitter and low packet loss probabilities, which require a fixed Quality-of-Service (QoS) assurance in transmission. The present Internet routing methods, based on the best-effort paradigm, is not able to provide any performance assurance required in these applications. Here is a need of the mechanism which will consider these QoS factors(delay ,jitter, bandwidth etc.) for the transmission. There are many aspects in the network which provide the guarantee of the quality of service. However the one of the key technology for providing it is the QoS routing. The basic problem of QoS routing is to find a path satisfying multiple constraints. It is focused on identifying the path that will consider multiple parameters like bandwidth, delay, jitter, cost, hop count etc. instead of one. To provide Quality of Service (QoS) guarantee both of the routing schemes –source routing and distributed routing can be used. In source routing, the path computation is done at source node whereas in distributed routing, the path computation is distributed among intermediate routers between source and destination. Both source routing and distributed routing have important roles to play in QoS routing. Source routing is seen impractical in Internet as the complete explicit path would have to be included in the IP header. In source routing, path computation is done at the source node. Whereas in distributed routing, path computation is distributed among set of intermediate nodes between source and destination. Source routing is used in today's Internet for special cases only, such as mapping the network with trace route, troubleshooting etc. Distributed routing is currently the dominant method in Internet. This paper describes the distributed routing approach and its implementation in QoS domain. Many distributed QoS routing algorithms have been proposed in literature by varying the QoS metrics and protocols. The paper discusses some of these algorithms and also provides their characteristics evaluation depicting its type, metrics considered and state information.

KEYWORDS: Distributed Routing, Quality of service, Distance vector routing, Link state routing, flooding

1. INTRODUCTION

QoS routing algorithms can be classified into two classes: source routing algorithms and distributed routing algorithms. They are classified according to how the state information is maintained and how the search of feasible paths is carried on. Both source routing and distributed routing have important roles to play in QoS routing.

In source routing, the source is required to be aware of the entire network topology. The route is predetermined and precisely specified by the source and is carried in a packet header. The route is specified in terms of the addresses of every router on the path from the source to the destination. The routers on the path only read the headers and usually cut-off or mark appropriate subfields in the header.

Source routing can be very flexible. Different packets with different QoS requirements can be sent on different routes. Because the routing decisions are done at a single place for each route, the routing algorithm can be very complex. It guarantees loop-free routes. Many source algorithms are conceptually simple and easy to implement, evaluate, debug, and upgrade. [1]

Source routing allows easier troubleshooting, improved trace route, and enables a node to discover all the possible routes to a host. It also allows a source to directly manage network performance by forcing packets to travel over one path to prevent congestion on another.

Source routing has not been widely adopted in the Internet. It is impractical for any single node to have access to detailed state information about all nodes and all links in a large network. Also, since the source node computes the whole path on its own, the computational overhead in the source node could be very high if the network is large. In general, source routing does not scale very well.

Source routing is used in today's Internet for special cases only, such as mapping the network with traceroute, troubleshooting, or forcing an alternate link to traffic flow to avoid congestion. Many QoS routing algorithms, however, are source routing algorithms [10]

In distributed routing, also called hop-by-hop routing, the paths are computed by distributed computation. Each router along the path will decide for the next hop to be taken by the packet. Here the packet carries just the destination address and each router looks into its routing table for the next hop to be taken for this particular destination. In addition, there need not be any routing-overhead in the packet, since all routing decisions can be made from the destination address alone. Its advantage is that the size of the header is less as compared to source routing.[1]

Distributed routing is more scalable than source routing. Problems arise if information in the nodes along the path is not consistent. This can cause routing loops and the path computation to fail. Distributed routing is the common strategy for routing in the Internet today.

Both routing models have significant position in QoS based routing. The distributed protocols will be the default solution, but source routing will be needed to override the default behavior of the network.

This paper realizes distributed routing approach to provide Quality of service. The paper is organized into 4 sections: Both of the source routing and distributed routing algorithms require to maintain a global state of the network at every node. The commonly used approach is to periodically exchange state information among all nodes by using either Distance vector protocols or link state protocols. Section II describes distance vector routing and link state routing approach. Section III discusses Quality of service. Section IV discusses various distributed QoS routing algorithms presented in literature and also provides their characteristics evaluation depicting its type, metrics considered and state information and section V concludes the paper.

2. DISTRIBUTED ROUTING IMPLEMENTATION

In Distributing routing every node participates in path calculation. The computational burden in a single node is much smaller than in source routing, since it only needs to find the next hop. Distributed QoS routing could be categorized into two types based on state maintained at each node. Maintenance of state information may be –

2.1 Local State:

Each node is assumed to maintain its up-to-date local state of all its QoS metrics like delay, the residual bandwidth, cost etc. to all its adjacent nodes. When the nodes are maintaining local state of QoS metrics, the most common approach to find the path is flooding.

2.2Global State:

The combination of the local states of all nodes is called a global state. Every node is able to maintain the global state by either a link-state protocol or a distance-vector protocol, which exchanges the local states among the nodes periodically.

2.2.1 Link State Approach

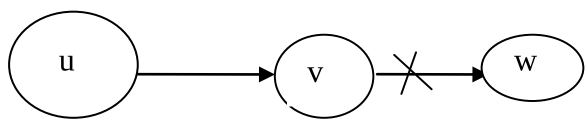
In this approach, the state of all local links is periodically broadcasted to all network nodes. Each node sends the information of its neighbours to every router. Each node of the network knows all the links and their states in the network. Based on this state, the required feasible path is locally determined. The nodes need to maintain an up-to-date version of the entire network topology at every node, which may constitute excessive storage and communication overhead[4].For distributed computation, link state routing utilizes distributed Dijkstra's algorithm.

Distributed Dijkstra algorithm is actually similar to Dijkstra algorithm. Each node floods periodically a control message throughout the network containing link state information. The algorithm will be same but the computation of the shortest path by each node is done separately based on link cost information received and available at a node. For this, entire topology knowledge must be maintained at each node.

2.2.2 Distance Vector Approach

In this approach, each node periodically sends the information, it has about the whole Internetwork, to its neighbours. The nodes of distance-vector protocols know only the distances or any other metric from their neighbours to the other nodes. Each node periodically exchanges distance vector information (distance and next hop from itself to all destinations) with its neighbors. Distance Vector protocols are simpler in nature.

Distance Vector protocols have several problems. The main problem with this approach is the lack of global knowledge, leading to problems such as slow convergence and routing loops. A routing loop is a path specified in the nodes routing tables at a particular point in time that visits the same node more than once before reaching the intended destination. Another problem is count to infinity. A node counts to infinity when it increments its distance to a destination until it reaches a predefined maximum distance value. E.g., consider two successive links (u, v) and (v, w) on a shortest path from u to the destination. The link (v, w) breaks i.e. its cost becomes infinite. Node u sends to v its estimate for its shortest path weight to destination and v thinks that u has a path with that cost to the destination. This however is not true since u's path was using broken link (v, w). However, u is not aware of the link break and v is not aware that u's path was (v, w). Node v performs update and sends to u its new estimate.



Node u uses this new estimate. The message exchange continues until the estimate reach the infinity. So, there has been a number of attempts to solve the counting-to-infinity and routing-table looping problems of distance-vector algorithms by increasing the amount of information exchanged among nodes [5].

In general, distance-vector protocols are simpler to configure and to understand than link-state protocols. They are also typically less processor intensive, freeing more of the router to other tasks, such as forwarding packets. Considering the message complexity, distance-vector routing protocols scale well to large network sizes, because each node (router) sends periodical topology update messages only to its direct neighbours. No flooding or broadcasting operations are involved. Each node maintains only limited information about the shortest paths to all other nodes in the network.

Distance-vector routing protocols are based on a distributed version of Bellman-Ford shortest path algorithm [6]. In Distributed Bellman-Ford algorithm, each node only needs to know the weights of edge incident to it, the identity of all the network nodes and estimates (received from its neighbours) of the distances to all network nodes. The classic

Bellman-Ford algorithm for calculating shortest paths can be easily adapted to a distributed environment in which the computations are performed locally by identical processors at each network node.

The algorithms based on global state work well in non-QoS networks but have some disadvantages for QoS networks. The periodical state exchange cannot guarantee the global state kept at each node up to date because the state of any node can change at any time. The global state kept by a node is always an approximation of the current network state due to the non- negligible delay of propagating local states. QoS routing is much more sensitive to the accuracy of the global state than non-QoS routing. Inaccuracy can make QoS fail. This algorithm may suffer from the same problem as with source routing, namely overhead in maintaining the global state and state impreciseness. Apart from degrading the routing algorithm's performance, the impreciseness can also create looping.

One of the mechanisms to avoid information imprecision is relying on local state. When local state for path computation is being used, no link state updating is required, as each node can check for its local states along the path. So, the overhead problem and inaccuracy of state information are eliminated in this case. If no global state is stored i.e. based on local state, techniques like flooding could be used to establish a path. Each node has to blindly flood the control messages in the network. This could again generate an overhead problem.

Selective flooding is a variation of flooding technique. Flooding based QoS routing means broadcasting the connection request parallel to all possible routes, performing QoS tests on all of them and finally selecting a better route among all the possibilities. The routing overhead in QoS flooding is less as compared to pure flooding concept because only that route will be followed which passes the QoS test. Thus in QoS flooding a request is flooded on the entire router's outgoing links (excluding the incoming link) which satisfy the QoS requirements of the request.

3. QUALITY OF SERVICE

Quality of Service(QoS) puts some restrictions in the form of certain constraints on the path. These constraints may be desired bandwidth, delay, variation in delay experienced by receiver(jitter),packet loss that can be tolerated, number of hops, cost of links etc.

The fundamental problem of routing in a network that provides QoS guarantee is to find a path between specified source and destination node pair that simultaneously satisfies multiple QoS parameters.

These parameters are represented in the form of metrics. One metric for each constraint is to be specified like bandwidth metric, jitter (variation in delay) metric, delay metric, number of hops metric, packet loss ratio etc. from one node to all other nodes in the network. Metric for a complete path with respect to each parameter is determined by the composition rules of metrics. The three basic rules are [11].-

(i) Additive Metric: The value of the constraint over the entire path is the addition of all links constituting path. For Example- delay, hop count, cost or jitter.

ii) Multiplicative Metric: Using this metric, the value for the complete path is multiplication of metric value of all its edges.

Examples are – reliability (1-lossratio) and error free Transmission (probability)

Multiplicative metric can be converted into additive by taking logarithm.

iii) Concave Metric: In this metric, either min edge value or max edge value is taken as constraint value for a path among all the edges of that path. For Example- Bandwidth

For a complete path, the constraints may be required either as a constrained form or in a optimization form. In constrained form, some condition is put on constraint value e.g. Choose that path only which has delay less than or equal to 60 ms. The path obeying the condition is called feasible. On the other hand optimization refers to path having minimum or maximum value for a constraint e.g. Choose the path that has minimum delay among all the paths. This path is called optimal path .

The further QoS issues have been discussed in[2][3].

4. DISTRIBUTED QoS ROUTING ALGORITHMS

Various distributed QoS algorithms have been proposed in literature by varying QoS metrics and maintenance of state. Some selected ones are listed here.

[1] has presented two distributed algorithms, one based on link state and other on distance vector. It has considered two metrics, delay and bandwidth. There must be an order in which the parameters of the routing metric are always evaluated, so that the second parameter is evaluated only for those paths considered best concerning the first parameter. It has been evaluated that delay is not as crucial to QoS based routing calculations as one might imagine. The propagation delay is likely to be far smaller than the queuing delay. Queuing delay, on the other hand, is highly dependent on the available bandwidth. The algorithm shall therefore compute the values for bandwidth first and then evaluate equal paths for their propagation delay. The resultant path is called a shortest-widest path. In order to find the optimal path, widest path has been found first. For this, it uses a modified Bellman-Ford algorithm using distance vectors.

The other algorithm is based on modified Dijkstra algorithm using link-states to find the widest path. If there is more than one path found, the algorithm chooses one with minimum length. That will be shortest path. Shortest paths can avoid loops, which widest paths lack.

[7] proposed a distributed heuristic approach to solve the DCLC(Delay-constrained least-cost) problem: the delay-constrained unicast routing (DCUR) algorithm. DCUR requires much fewer messages, and therefore, it scales well to large network sizes. DCUR requires only a limited amount of information at each node. This information is stored in a delay vector and a cost vector that are similar to the distance vectors of some existing routing protocols. The basic idea of DCUR is to restrict the amount of computation by limiting the number of links to choose from when constructing a path for a given source-destination pair. When a node receives a request to construct a delay-constrained path to a given destination, that node is given the choice between two alternatives only. The node can either follow the direction of the LC (least-cost)path or the direction of the LD(least-

delay) path. After deciding which direction to follow, the node sends a request to the next hop node in that direction to take over responsibility for the rest of the path construction operation. When the next hop node receives a request to construct a delay-constrained path, it follows the same procedure that has been followed by the previous node. Each node gives the responsibility for path construction to the next hop node in the direction of the destination until the destination itself is reached. Limiting the number of paths to choose from at any node to only two restricts the amount of computation considerably. DCUR is always capable of constructing a delay-constrained path within a finite time, if one exists, for a given source-destination pair. DCUR is capable of detecting and eliminating any loops that may occur while it constructs a delay-constrained path. A loop is detected if the control message visits a node twice. Whenever it happens, the routing process is rolled back until reaching a node from which the least-cost path was followed. The routing process resumes from there by changing the next hop along the least-delay path. It was proved that such a mechanism removes all loops.

[8] has considered the same DCUR algorithm as in [7] and improved the original DCUR by reducing the communication complexity. This new efficient distributed routing algorithm, which can easily compute a loop-free delay-constrained path with reduced message complexity, requires only a small amount of computation at any node. The algorithm can always find a delay-constrained path if such a path exists. At each network node, limited information about the network state is needed .The proposed algorithm improves the worst-case performance of [7] algorithm by avoiding loops instead of detecting and removing loops. A control message is sent to construct the routing path. The message travels along the least-delay path until reaching a node from which the delay of the least-cost path satisfies the delay constraint. From that node on, the message travels along the least-cost path all the way to the destination.

[9] proposed a multi-constrained distributed QoS routing scheme, which does not require state information distribution and complex route computations. This is based on flooding concept and relies on local link state information. The proposed algorithm has built-in quick pruning mechanism that is crucial for controlling the 'over-reservation' problem,

and has absolutely no implementation problems unlike the other flooding-based QoS routing schemes. The routing overhead can be significantly reduced, as the request packets are allowed to take only those routes passing the QoS admission test. The established connections will guarantee all the QoS metrics specified (bandwidth, delay and delay jitter etc.).

In [13], Shin and Chou proposed a distributed routing scheme for establishing delay-constrained connections. It guarantees to find a qualified route by searching in parallel. No global state is required to be maintained at any node. The algorithm floods routing messages from the source towards the destination. In order to reduce the number of request message, it prunes infeasible routes quickly, thereby reducing the operational overhead. Each message accumulates the total delay of the path it has traversed so far. When an intermediate node receives a routing message, the message is forwarded only when one of the following two conditions are satisfied. One-It is the first such message received by the node, and second- it carries a better-accumulated delay than the previously received messages. If either condition is true, the message will be forwarded along the outgoing links whose delay plus the message's accumulated delay does not exceed the end-to-end delay requirement. Once a message reaches the destination, it finds a delay-constrained path, which is the one it has traversed.

[14] presents a ticket-based distributed QoS routing algorithm. The main objective of this algorithm is to handle the imprecise state information. To achieve its purpose, initially, Chen and Nahrstedt suggests a simple imprecise state model that defines which information must be stored in every node: connectivity information, delay information, cost information and an additional state variable named delay variation, which stands for the estimated maximum change of the delay information before receiving the next updating message. For simplicity reasons, the imprecise model is not applied to the connectivity and cost information. They justify this assumption by saying that the global routing performance is not significantly degraded. Then, a multipath distributed routing scheme, named ticket based probing is proposed. The ticket based probing sends routing messages, named probes, from a source to a destination. Based on the (imprecise)

network state information available at the intermediate nodes, these probes are routed on a low-cost path that fulfils the delay requirements of the route request. Each probe carries at least one ticket in such a way that by limiting the number of tickets, the numbers of probes are limited as well. Moreover, since each probe searches a path, the number of tickets also limits the number of searched paths. In this way, the trade-off between the signaling overhead and the global routing performance may be controlled. Finally, based on this ticket based probing scheme, Chen and Nahrstedt suggests a routing algorithm to address the NP-complete delay-constrained least-cost routing problem, called Ticket Based Probing algorithm.

In [15], Kweon and Shin suggest a technique of bounded flooding. To reduce the overhead in flooding, along with the QoS constraints an additional constraint is imposed on the number of hops the connection request can travel. Thus, according to this algorithm, routing messages are flooded into the network for a certain hop count in all directions. No global state is required to be maintained at any node. This approach can be interpreted as flooding with a limited search area. In order to limit the search area, a distance table at each node is maintained which stores hop counts of the minimum hop paths to all nodes reachable via its outgoing links. A qualified route for each requested real time connection is searched by flooding request message with a limited hop count. However, it is possible that with certain sets of constraints, the hop count will not include the optimal path in the search space.

The approach proposed in [16] is based on flooding approach but it uses additional state information and reduces the overhead in connection establishment. In the proposed algorithm, router stores information about its immediate neighbours (routers reachable in one hop) and second-degree neighbours (neighbours of a neighbour). The advantage of this approach is twofold. First, the message overhead and the impreciseness will not be as large as maintaining the global state. A router exchanges information only with its neighbours. As a result, the impreciseness in storing the information about the second-degree neighbours will not be as big as the impreciseness in storing the entire global state. Second, using the information about the second-degree neighbour, a router can forward the connection requests intelligently instead of blindly flooding the requests. This is because every router can now see two levels downstream. Hence, the overhead in connection establishment is reduced.

[12] proposed a family of distributed routing algorithms. These algorithms require every node to maintain local state. The algorithms use a distributed computation to collectively utilize the most up –to-date local information at each node to find a path. The found path is guaranteed to be loop free. After a connection request arrives, probes are flooded selectively along those paths, which satisfy the QoS and optimization requirements. Unlike in [15] probes are forwarded only to a subset of outgoing links selected based on the topological distances to the destination. Further, the probes proceed only when the nodes and the links on the way have sufficient resources. This reduces the overhead as compared to [15].

CHARACTERISTIC EVALUATION

A characteristic evaluation of Distributed QoS routing algorithms discussed in this chapter has been presented in the following table:

Table1: Characteristic Evaluation of Distributed QoS Routing Algorithms

Algorithm	**QoS Metrics**	**State for path Calculation**	**Type of Information**
Wang-Crowcroft [1]	Bandwidth, Delay	Global	Link State
Wang-Crowcroft [1]	Bandwidth, Delay	Global	Distance Vector
Salama et al. [7]	Delay, Cost	Global	Distance Vector
Sun-Landgendorfer[8]	Delay, Cost	Global	Distance Vector
Song-Pung et al.[9]	Generic	Local	Link State
Shin-Chou-Kweon [13]	Delay, Hop Count	Local	Link State
Chen-Nahrstedt	Generic	Global	Distance

Ticket based Routing[14]			Vector /Link State
Kwean-Shin [15]	Bandwidth, hop count	Local	Distance Vector
Ghosh, Sarangan, Acharya [16]	Generic	Partial Global	Distance Vector
Chen-Nahrstedt (DRA) [12]	Generic	Local	Links State/ Distance Vector

5. CONCLUSION

To provide Quality of Service in the current scenario of Internet is a key problem in computer network research. There are many aspects which networks provide the guarantee of the quality of service. However the one of the key technology for providing the guarantee of the Quality of Service is the QoS routing. Both of the routing schemes – source routing and distributed routing can be used to provide QoS routing. In source routing, the path computation is done at source node. Source routing has not been widely adopted in the Internet. It is impractical for any single node to have access to detailed state information about all nodes and all links in a large network. Source routing is used in today's Internet for special cases only, such as mapping the network with trace route and troubleshooting. In distributed routing, the path computation is distributed among intermediate routers between source and destination. Distributed routing is more scalable than source routing. Today distributed routing is the common strategy for routing in the Internet. Thus, in this paper, we have discussed distributed QoS routing approach.

In order to support QoS routing, network nodes require accurate state information about the available resources. State information can be local state or global state. Each node is assumed to maintain its local state information up-to-date e.g. Node delay, the residual bandwidth of the outgoing link, available buffer size etc. The union of all nodes' local states is called the global state. The global state kept by a node is always an approximation of the current network due to the delay of propagating local states with the growing network size. This global state is maintained in every node by exchanging the local states between the nodes. There are two popular ways of doing this -Distance Vector Protocol and Link State Protocol.

The most common approach in source routing to maintain global state is link state protocol. Distributed routing can be implemented by either using local state or global state. Both of the link state protocol and distance vector protocol are used in maintaining a complete global state at every node. However, in implementation, the global state information available for making the routing decisions at each node is often inaccurate in a lively environment. The routing algorithm does not provide satisfactory performance with imprecise state information. QoS routing is much more sensitive than non-QoS routing in terms of the accuracy of the global state. Inaccuracy can lead QoS to failure. Therefore, the design of routing algorithms for large networks should take the information imprecision into consideration.

One of the mechanisms to avoid information imprecision is relying on local state. As far as path computation strategy is concerned, techniques like flooding can be used to establish a path. Many distributed QoS algorithms have been proposed in literature by varying the QoS metrics and state of information maintenance. This paper has presented the characteristic evaluation of distributed QoS routing algorithms. An important contribution in these works is using local state as the global state frequently changes [12]. Our future work will focus on it.

REFERENCES

1. Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," IEEE J. Select. Areas Commun., vol. 14, pp. 1228–1234, Sept. 1996.

2. S.Upadhaya, G.Devi , "Exploring Issues for QoS Based Routing Algorithms" published in International Journal on Computer Science and Engineering (E-ISSN : 0975-3397) Vol. 02, No. 05, 2010, 1792-1795

3. . S.Upadhaya, G.Devi , "Characterization of QoS Based Routing Algorithms" , International Journal of Computer Science & Emerging Technologies 133 Volume 1, Issue 3, October 2010

4. J. Garcia-Luna-Aceves, "A unified approach to loop-free routing using distance vectors or link states," in Proc. SIGCOMM'89, Austin, TX,Sept. 1989

5. J. Baras, G. Theodorakopoulos, J. Walran Path Problems in Networks Synthesis lectures on communication networks,Morgan & Claypool Publishers, 2010

6. David Walden. The Bellman-Ford Algorithm and ”Distributed Bellman-Ford”. http://www.walden-family.com/public/bf-history.pdf.

7. Salama, H. F., Reeves, D. S. & Viniotis, Y.(1997). A Distributed Algorithm for Delay-Constrained Unicast Routing, INFOCOM’97, Japan, April 1997.

8. Q. Sun, H. LangendGrfer,” A new distributed routing algorithm for supporting delay-sensitive applications”, Computer Communications 21 (1998) 572-578

9. J. SONG, H. K. PUNG and L. Jacob,” A Multi-Constrained Distributed QoS Routing Algorithm”, 7695-0777, 2000 IEEE

10. S. Chen , K. Nahrstedt, An overview of quality-of-service routing for the next generation high-speed networks: Problems and solutions, IEEE Network, vol. 12, no. 6, pp. 64–79, Nov-Dec 1998.

11. Xiao, X. & Ni, L. M.(1999). Internet QoS: A big picture, IEEE Network, vol.13, no. 2, pp. 8–18, March-April 1999.

12. Chen, S. & Nahrstedt, K. (1998). Distributed Quality-of-Service Routing in High-Speed networks Based on Selective Probing, Tech. rep., Univ. of IL at Urbana-Champaign, Dept. comp. Sci., 1998.

13. Shin, K. G., Chou, C. C. & Kweon, S.K.(1995). A Distributed Route Selection for Establishing Real-Time Channels, Proc. Of 6-th IFIP Int’l Conf. on High Performance Networking pp.319-329, Sept. 1995.

14. Chen, S. & Nahrstedt, K. (1998). Distributed QoS Routing with Imprecise State Information, Proc. Of ICCCN’98,Louisiana, Oct. 1998.

15. Kweon, S. K. & Shin, K. G.(1999). Distributed QoS Routing Using Bounded Flooding, University of Michigan CSE technical report, CES-TR-388-99.

16. Ghosh,D., Sarangan,V. & Acharya,R. (2001) Quality of Service Routing in IP Networks, IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 3, NO. 2, JUNE 2001.

