ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 8(20), DOI:10.17485/ijst/2015/v8i20/49253 August 2015 ISSN (Online) : 0974-5645

An Approach to Distributed Multi-Path QoS Routing

Gaytri Devi[1*] and Shuchita Upadhyaya[2]

[1]GVM Institute of Technology and Management, DCRUST, Sonipat - 131001, Haryana, India; gayatri.dhingra1@gmail.com

[2]Department of Computer Science and Applications, Kurukshetra University, Kurukshetra - 136119, Haryana, India; shuchita_bhasin@yahoo.com

Abstract

The principal objective of the present research work is to attain QoS in routing mechanism. For this, a distributed QoS multi-path routing algorithm has been designed that finds multiple disjoint paths in a distributed way from one source to one destination. These multiple paths satisfy the given QoS requirements. The proposed algorithm DQM (Distributed QoS Multi-Path algorithm) uses local information to find multiple QoS paths as the global state information available for making routing decisions is often inaccurate in a lively environment and inaccuracy can be the cause of QoS failure. The QoS parameters that have been considered are residual bandwidth and delay. Firstly the algorithm obtains all the paths which satisfy the QoS criteria in a fully distributed fashion. At destination, these paths are again processed to identify mutually exclusive disjoint paths. Among overlapping paths, those paths will be selected whose combination of bandwidth and delay is better than other. To achieve this, fuzzy logic has been used. By applying fuzzy logic a new single metric is obtained from the input's bandwidth and delay of the calculated paths. Based on this value, the paths are filtered to disjoint paths. Since the paths obtained at destination are already constrained paths and further these paths are refined into most optimal disjoint paths using the concept of fuzzy logic, thus the proposed strategy provides both constrained and optimal paths.

Keywords: Bandwidth, Delay, Distributed Routing, Fuzzy, Local State, Multi-path, Quality of Service

1. Introduction

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

- Additive Metric: The value of the constraint over the entire path is the addition of all link values constituting path. For Example- delay, hop count, cost, jitter etc.
- 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.

- 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 an 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

** Author for correspondence*

is called feasible path. 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[1,2].

Multipath routing strategy can be used for the transmission of QoS sensitive traffic over the network. Multipath routing establishes multiple routes between source and destination nodes. The purpose of having multiple paths is either enhancing the reliability of the data transmission or allowing load balancing.

If multiple paths are being used for the transmission of the traffic, then the traffic will be redirected to the backup path if active path fails. In this way, robustness can be achieved. Load balancing in a network can be achieved by splitting the traffic between a given source and destination across multiple paths. So multipath can be proved very valuable for the efficient implementation of Quality of Service.

For multiple paths, there are two characteristics to address: path quantity and path independence. Path quantity means number of paths and path independence refers to disjoint paths. Disjoint paths may be either link-disjoint or node-disjoint. A pair of paths is link disjoint if, they have no common (i.e. overlapping) links. A pair of paths is node-disjoint if, they have no common nodes. One can address the issues of load sharing and fault-tolerance only if the routes satisfy some form of disjointness[3]. The present strategy has also been focused on finding multiple disjoint paths.

To provide Quality of Service (QoS) guarantee both of the routing schemes – source routing and distributed routing can be used. Both source routing and distributed routing have significant roles to play in QoS routing. Source routing is used in today's Internet for only special cases only like troubleshooting. Distributed routing is the dominating strategy for routing in the Internet.

In order to support QoS routing, network nodes require accurate state information about the available resources i.e. QoS values. 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 local states is called the global state. The global state kept by a node is always an approximation of the current network.

Distributed routing can be implemented by either using local state or global state. 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 make QoS failure. Therefore, the design of routing algorithms for large networks should take the information imprecision into consideration.

In this situation, we can rely on local information maintained by each node. When we implement distributed routing with local information, most often flooding-based algorithms are used.

One such algorithm is DRA proposed by[4] that has considered information imprecision to calculate QoS paths in a distributed way. Based on this algorithm, this paper proposes a distributed QoS multiple path searching algorithm - DQM. For the purpose of understanding DRA approach, more clearly, an illustration has been described in[5].

The proposed algorithm DQM (Distributed QoS Multi-Path algorithm) searches multiple paths satisfying QoS requirement, reaching from one source to one destination within a network, without loops. Firstly the algorithm obtains all the paths that satisfy the QoS criteria from a source to a destination in a fully distributed fashion using only local information of resources. Here, the QoS criteria include – residual bandwidth and delay. Then at destination, these paths are filtered to the mutually exclusive disjoint paths by applying fuzzy logic. The fuzzy logic has been applied to the two QoS parameters of obtained paths to get one QoS value in order to incorporate the magnitude of both of the parameters. Those disjoint paths are selected whose QoS value is stronger than others. Fuzzy approach has been used to accommodate the fair goodness of both of the metrics-bandwidth and delay for identifying exclusive paths.

Thus the proposed strategy will provide both constrained and optimized paths. This is because the strategy is able to provide routing paths which are constrained with residual bandwidth and delay. Also, the strategy is providing optimal disjoint paths from these constrained paths. The proposed strategy has been verified manually for one example network and has also been presented in this paper.

The paper first provides an overview of the proposed approach along with its algorithm in section 2. Subsequently in section 3, an illustration of the path calculation algorithm is given through an example network and also, the strategy of finding disjoint paths out of the obtained paths using fuzzy logic is explained along with its illustration. Section 4 concludes the paper and presents the future work.

2. Distributed QoS Multi-Path Approach (DQM)

2.1 Overview of the Algorithm

The proposed approach is based on DRA algorithm for packet –switching network. DRA is a distributed algorithm that utilizes single path between a source-destination pair and reserve resources on that path.

An expansion of DRA is proposed in the DQM algorithm by identifying multiple paths between a source – destination pair. However the resource reservation strategy will remain the same as in case of DRA.

DQM algorithm searches multiple paths satisfying QoS requirement between a source-destination pair within a network. Though the message flooding has been used for path searching, it reduces control traffic by adopting selective flooding. In selective flooding the routers don't send every incoming packet on every outgoing line; only those lines are chosen which are going approximately in the right direction. Selective flooding has much less overhead as the packets are flooded only over links which are anticipated to lie on a path to the destination. Selective flooding requires storing the network topology. Such topology information is fairly static and need not be updated on a regular basis.

To implement this, we have assumed a selected domain of the network where the algorithm will operate and all the links chosen from the source are the ones that lead to destination. In addition to it, the messages are sent only on those paths which satisfy the Quality of Service requirement, consequently, restricting the paths for message flow.

This fully distributed approach uses local information of QoS parameters, stored at each node and is independent of the global information to find the path. Every node is required to maintain the delay and residual bandwidth of its all outgoing links. Assumption has also been made that each node in the network is maintaining the topology of the network either by using distance vector or link-state protocol.

The DQM algorithm uses two types of messages – PROBE and ACK. The PROBE message finds the multiple paths and ACK message provides the path information back to the source. Source starts the process of message transmission towards destination by sending the PROBE message to its neighbours which satisfy QoS requirement. Each node, after receiving the PROBE message, adds itself to the path list, compares the bandwidth, increases the hop count, adds the delay, creates a new PROBE message with proper values and then sends it to all neighbouring nodes that are not listed on the path list provided they meet the QoS criteria. This guarantees to generate loop-free and QoS constrained paths.

Only those multiple paths will be obtained at the destination which satisfy the given bandwidth and delay requirement. At destination these paths are filtered to mutually exclusive link disjoint paths.

The strategy that has been used for finding disjoint paths is fuzzy logic.

Fuzzy logic is the new emerging technology for evaluating the efficiency of various systems. Fuzzy logic is tolerant in imprecise data, nonlinear functions and can be mixed with other techniques for different problems solving[6].

Since all the paths are satisfying bandwidth and delay constraints, from the path list, disjoint paths can be chosen by considering either metric - bandwidth or delay. When we choose the disjoint path according to the bandwidth then among the paths having overlapping edges, the path of greater bandwidth can be selected in disjoint list. In this way the delay value will be ignored.

This can be done in other way i.e. choosing the disjoint paths based on delay metric (giving preference to paths having lesser delay) consequently ignoring bandwidth value.

In order to accommodate the strengths of both of the metrics in identifying the most QoS support paths, the two parameters can be considered together by combing them into one using Fuzzy logic. By introducing fuzzy logic, multiple QoS parameters can be combined. The proposed approach introduces a fuzzy metric i.e. FM metric produced from bandwidth and delay metrics of found paths .This FM metric represents QoS path significance value. In this way we will be able to choose more appropriate QoS significant paths. The major processes to design a Fuzzy Logic System (FLS) are - Fuzzification, Rule-Base, Fuzzy Inference and Defuzzification. Fuzzy logic system brings about the creation of fuzzy sets , membership functions corresponding to input -output linguistic variables and a rule matrix which will define a rule base upon which the decision will be based. After defining fuzzy rules and membership functions, fuzzy inferenceing is performed to reach the output. Then defuzzification combines all the output values to final conclusion[7].

The concept of fuzzy logic has also been introduced in[8,9] where routing model based on fuzzy mixed metric has been proposed. Fuzzy value corresponding to each link has been calculated. Based on this fuzzy value, the best path has been found. As the defuzzification has not been performed, so minimum value of linguistic term has been considered as best value for the link. Delay and load were considered as QoS parameters. But here, in DQM approach, fuzzy values of complete path are calculated and compared. Defuzzification is also applied to get one output value that will represent the quality of the path.

2.2 Design of the Algorithm

A network is modelled as a set V of nodes that are interconnected by a set E of full-duplex, communications links. The proposed approach is based on QoS metrics residual bandwidth, delay and hop count.

The delay and bandwidth of a path p = S->i->j->....->k->D are defined as follows

Delay (p) = delay (S, i) + delay (i, j) +....+ delay (k, D)

Bandwidth (p) = min { bandwidth (S, i), bandwidth (i, j) ....bandwidth (k, D)}

Hop count (p) = n i.e. number of hops taken by PROBE message to find path p.

Here i, j, k are intermediate nodes

S-Source node

D-Destination node

Information Stored At Each Node

At each node, the following information is required to be stored.

The local information to be stored at each node i

$N_{i,j}$ -List of adjacent nodes of i , to which data packets for destination may be forwarded.

$B_{i,j}$- Residual bandwidth of outgoing links from i.

$D_{i,j}$- Delay of outgoing links from i

Here j varies from 1 to total number of neighbour nodes.

$B_{i,j}$ and **$D_{i,j}$** are given with the network.

Status Table

Since multiple paths are being calculated, the status table stores multiple values relative to each path. The status table contains the following information related to identified path at each intermediate node and at destination as well-

Cid- Connection id

$Predecessor_{i,k}$- It stores predecessor of k^{th} path for node i.

$Bandwidth_{i,k}$ –It stores the bandwidth of the k^{th} path traversed till now for node i.

$Delay_{i,k}$ – It stores the total delay of k^{th} path from source to node i.

$Hops_{i,k}$ =It stores the total number of hops of k^{th} path from source to that node i.

Path [j] $_{i,k}$ - It stores the intermediate nodes of k^{th} path from source to that node i. j represents the nodes of traversed path so far.

Here k varies from 1 to total number of paths identified at that node.

The Format of Messages

The information in the PROBE packet includes the connection identification, source, destination, sender of message, the QoS requirement, path and the current QoS status.

The format of the PROBE message has been shown in the Table 1.

Table 1. PROBE Message Format

T	S	D	K	Cid	BC	DC	Path	Bws	Dls	Hops

The fields of Probe message represent the following values-

T- Indicate the type of packet (PROBE or ACK)

S - Source node

D - Destination node

K- Sender of the packet.

Cid- Connection identification is the unique identification code assigned by the source to represent routing request.

BC, DC - Represents QoS requirement.

BC- Represents the bandwidth constraint or bandwidth requirement .The residual bandwidth of each edge of the identified path should not be less than BC.

DC- Represents the delay constraint. The delay value of the identified path(sum of delay of all the edges of path) should not be greater than DC.

Path - List of intermediate node Ids that the message has traversed so far.

Bws, Dls, Hops –Present the QoS status.

Bws- Contains the bottleneck bandwidth of the path traversed so far (the minimum bandwidth encountered

for traversed edges) as the bandwidth is concave metric.
Dls- Contains the accumulated delay that passes through the path (addition of delay of all the edges traversed so far) as the delay is additive metric.
Hops- Contains the number of hops traversed so far.
Format for ACK message is given in Table 2. The fields of Ack message have been explained here.

Table 2. ACK Message Format

T	S	D	K	Cid	BC	DC	Dj_path	FM

T- Indicates the type of packet.
S - Source node.
D - Destination node.
K- Sender of the packet.
Cid - Connection identification is the unique identification number assigned to route request by the source.
BC- Represents the bandwidth constraint.
DC- Represents the delay constraint.
Dj_path - Contains nodes of disjoint paths.
FM- Fuzzy metric value for that disjoint path.

[BC & DC fields will be returned back for reserving resources on the path. However reservation of resources has not been illustrated in the proposed algorithm].

2.3 DQM Algorithm

As discussed above, DQM algorithm includes finding multiple paths and selection of disjoint paths. The algorithm has been explored here

2.3.1 Finding Multiple QoS Paths

For calculation of multiple QoS paths, different nodes perform different operations. The operations performed at each node have been explored below.

2.3.1.1 At Source Node

Source will start the process by sending the PROBE message to all its adjacent neighbours which satisfy the QoS requirement. A timeout period will be set for the ACK to be received at source from the destination. Once the source node receives acknowledgement from the destination node, it will have the routing information with QoS guarantee for transporting data packets.

2.3.1.2 At Intermediate Node

At the intermediate nodes, each adjacent edge is checked for the required QoS satisfaction. It will forward the PROBE to all its neighbours that are satisfying the QoS requirement by altering the fields Dls, Bws, Hops, Path. Certain values of the PROBE will also be recorded in the intermediate node. The process can be described as follows-

Step 1-- The fields- Cid, Bws, Dls, Hops, K and Path of PROBE will be stored in the status table of the current node.

Step 2--The current node will check all its adjacent nodes and selects only those nodes, which are not in the path list of the PROBE to forward it so that loops can be avoided.

Step 3-- It will calculate the corresponding value of the bandwidth and delay of all adjacent edges by looking into its local information table. If bandwidth value is less than required bandwidth or delay accumulated is greater than the delay threshold for all the neighbours, the PROBE will be discarded. Otherwise it will update the message for all its qualifying neighbours as

- Choose the minimum bandwidth among the bandwidth specified in the message and bandwidth of the edge going to be traversed.
- Add the delay of the edge to be traversed in the delay specified in the message.
- Increment the value of Hops.
- Add name of intermediate node in the path list.

This alteration will be done individually for all its adjacent nodes which are satisfying QoS criteria and have passed the test on step 2 .The other values of the message will remain same.

2.3.1.3 At Destination Node

Upon receiving a first PROBE of a particular connection ID, the destination node records the delay value, Cid, sender, bandwidth, the source node and the complete path in the status table and waits for time period until the others arrive.

After receiving all the paths, the destination node stores all the PROBES in its status table, calls the disjoint path selection procedure to find disjoint paths. Once the destination node will have the disjoint paths, it will send the ACK message on only disjoint paths.

The DQM algorithm has been outlined in Table 3.

Table 3. DQM algorithm

```
DQM Algorithm
/*initialize variables*/
{At source node}
{case1: Start Process}
/*Source will start the process by sending a probe message to all its neighbours for the given destination which qualify QoS requirement.*/
For every node j belongs to S. neighbour; do
If condition (S; j; QoS: bandwidth –BC, delay-DC): B (S, j) >= BC and D (S, j) <DC is true then
Bws=B[S,j];
Dls=D[S,j];
Hops=1;
Path=S;
Send j a PROBE[S; T; S; Cid; BC; DC; Path []; Bws; Dls; Hops]
End if
End for
{Case 2 : Receieve  ACK}
The connection can be established
{At intermediate node n_i}
{Case1: Receieve PROBE}
Record the information (Cid, predecessor, bandwidth, delay, hops, and path) at the node in the form of status table .
For every node j belongs to i. neighbour; j ≠ path[] do
/* j does not belong to path list of the probe message*/
If condition (i; j; Qos: bandwidth –BC, delay-DC): B (i,j) >= BC and D (i, j) + Dls (p) <DC is true then
/* update the fields */
Bws = min(Bws, B[i,j]);
Dls=Dls + D (i, j)
Hops = Hops+1;
Add i to path[] /* add i to the path list */
Send j a updated PROBE [S;T; i; Cid; BC; DC; path []; Bws; Dls; Hops]
Endif
Endfor
{Case 2: ACK receieved} Forward  the probe to the predecessor [of that path]
At destination
Case 1: PROBE message received}
T will wait for a time stamp value to receive all the multiple paths if the time slag is over and destination has recorded the paths then fuzzy_disjoint() /* call function to find disjoint paths*/
End if
{Case 2 : start ACK message}
Send the ACK message to all the predecessor of only disjoint paths.
```

2.3.2 Disjoint Path Selection Procedure

To find the disjoint paths, fuzzy disjoint path procedure is supposed to maintain two lists –path list and disjoint list. The procedure to find disjoint paths i.e. Fuzzy_disjoint path algorithm is as follows-

Step 1 - Calculate the QoS path significance value FM (Fuzzy Metric) value by applying fuzzy logic i.e. Fuzzification, Fuzzy Inference, De-fuzzification on all the paths calculated .

Step 2 -Sort the paths in increasing order based on the FM in path list [the least FM value is the best value]

Step 3 - From path list move the first path to disjoint list.

Step 4 - Compare the edges of the next path in the path list with the paths available in disjoint list. if, this path have similar edges with any of the path present in disjoint list, then this path will be discarded as the path having lesser value has to be chosen for disjoint list and on the occurrence of a tie the path whose hop-count value is less will be selected. If, this path has no overlapping edges with the path available in disjoint list then the path will be added to disjoint list.

Step 5 - If there is no path in path list, terminate the process. Otherwise, go to step 4.

3. Example Network

The illustration of proposed algorithm including path calculation and disjoint path selection with the example network has been presented here.

3. 1 Illustration of the Path Calculation

As discussed above, each node keeps the up to date local state about all outgoing links. The state information includes the residual bandwidth, delay of all its neighbours. For illustration assumption of these values has been taken. Here after wherever the word bandwidth is used, it will be representing residual bandwidth. The network is of 6 nodes. In this network, each link is labelled by residual bandwidth and delay. For simplification, delay and bandwidth value from i to j and j to i has been taken identical. The Example Network has been shown in Figure 1.

Every node maintains the state of local information i.e. residual bandwidth and delay of its adjacent nodes in data structure $\mathbf{N}_{i,j}$ $\mathbf{B}_{i,j}$, $\mathbf{D}_{i,j}$ as shown in Tables 4-9.

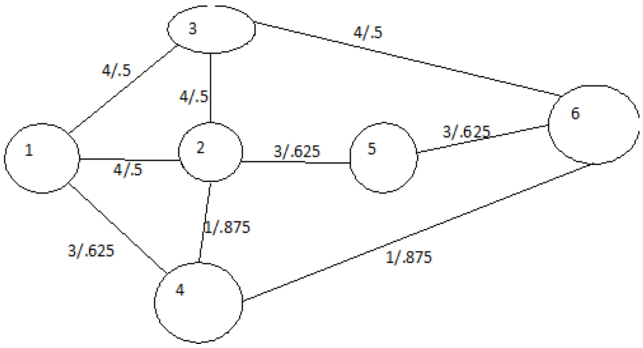

Figure 1. Network with edges depicting residual bandwidth/delay.

Table 4. Node 1

Node	Bandwidth	Delay
2	4	.5
3	4	.5
4	3	.625

Table 5. Node 2

Node	Bandwidth	Delay
1	4	.5
3	4	.5
4	1	.875
5	3	.625

Table 6. Node 3

Node	Bandwidth	Delay
1	4	.5
2	4	.5
6	4	.5

Table 7. Node 4

Node	Bandwidth	Delay
1	3	.625
2	1	.875
6	1	.875

Table 8. Node 5

Node	Bandwidth	Delay
2	3	.625
6	3	.625

Table 9. Node 6

Node	Bandwidth	Delay
3	4	.5
4	1	.875
5	3	.625

A new QoS request flow of cid A10, originates as source node 1 for destination 6 with bandwidth constraint 3 and delay constraint 4 unit of time. The objective is to identify multiple mutually exclusive paths that satisfy QoS requirement of the flow between node 1 and 6. The working of the proposed scheme is depicted in the form of stages that trace the status of the routing procedure followed on every level hop. At every node, the message forwarding and information updating has been shown as below.

Stage 1

Node 1

Node- 1 starts finding paths by sending a message to its neighbours which satisfy the QoS requirement. Here 2, 3 and 4 all fulfil the requirement as their bandwidth is greater than/equal to 3 & accumulated delay is less than 4. According to the message format of PROBE, the field's values have been identified for the edge 2 as PROBE {1, 6, 1, A10, 3, 4, 1, 4, .5, 1}

Bws is taken as 4 (bandwidth value from 1 to 2). Since it is first edge to be traversed from source, bandwidth value from 1 to 2 will be considered as bandwidth of the path.

Dls is taken as .5 (The delay value from 1 to 2).Since it is the first edge to be traversed from source, delay value from 1 to 2 will be considered as accumulated delay.

The other fields are self-explanatory.

Thus, 1 sends the PROBE message along the edges 2 as-

PROBE {1, 6, 1, A10, 3, 4, 1, 4, .5, 1}

Similarly 1 sends the PROBE message along the edge 3 as - ROBE {1, 6, 1, A10, 3, 4, 1, 4, .5, 1} and along the edge 4 as - PROBE {1, 6, 1, A10, 3, 4, 1, 3, .625, 1}. This message sending process has been shown in Figure 2.

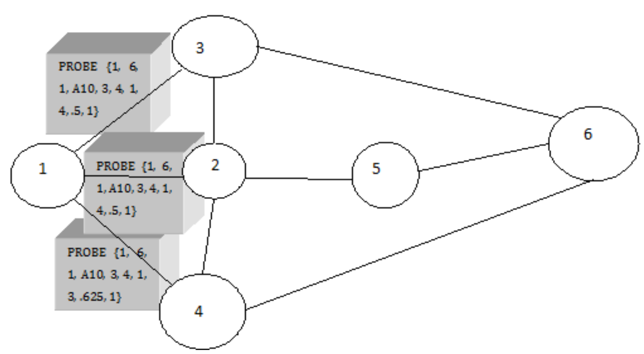

Figure 2. PROBE messages sent by Node 1.

After receiving the probes, Node 2, 3 & 4 stores the information of probes in their respective status table as Tables -10, 11 and 12.

Table 10. Status Table of Node 2

Predecessor	Cid	Path	Bandwidth	Delay	Hops
1	A10	1	4	.5	1

Table 11. Status Table of Node 3

Predecessor	Cid	Path	Bandwidth	Delay	Hops
1	A10	1	4	.5	1

Table 12. Status Table of Node 4

Predecessor	Cid	Path	Bandwidth	Delay	Hops
1	A10	1	3	.625	1

Stage 2

Node 3

Now node 3 has three adjacent edges 1, 2 and 6. As shown in Figure 3, Node 3 updates the fields in the message and forwards the message to 2 and 6 as both edges are satisfying bandwidth and delay criteria. It will not forward the message to 1 to avoid the loop because 1 has been included in the path.

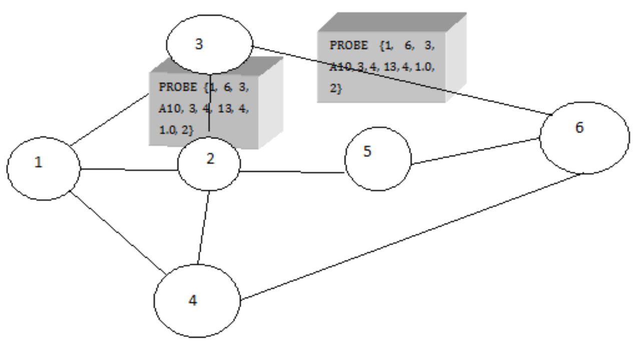

Figure 3. PROBE message sent by Node 3.

For node 2, the updation of bandwidth and delay in the message will be done as follows

The updated bandwidth is 4 (minimum value between bandwidth of message and bandwidth of edge 3-2) and accumulated delay is 1.0 (sum of delay specified in message i.e. 5 and delay of 3-2 edge i.e. .5).

Both are satisfying the constraints as bandwidth constraint is 3 and delay constraint is 4.

Thus, for the node 2, the PROBE message is-
PROBE {1, 6, 3, A10, 3, 4, 13, 4, 1.0, 2}
Similarly
For the node 6, the PROBE message is
PROBE {1, 6, 3, A10, 3, 4, 13, 4, 1.0, 2}

6, destination receives the first path and stores the path in its status table as shown in Table 13.

Table 13. Status Table of Node 6

Predecessor	Cid	Path	Bandwidth	Delay	Hops
3	A10	13	4	1.0	2

2 will receive another message from 3. It will keep this information in its table as Table 14.

Table 14. Status Table of Node 2

Predecessor	Cid	Path	Bandwidth	Delay	Hops
1	A10	1	4	.5	1
3	A10	13	4	1.0	2

Node 2

Node 2 has four adjacent edges - 1, 3, 4 and 5. 2 will update the message (received from 1) and forward the message to 3 and 5 only as shown in Figure 4. Node 4 does not satisfy bandwidth requirement. The edge 2-4 is having bandwidth 1 and the bandwidth constraint is 3. So, it will not send PROBE to 4.

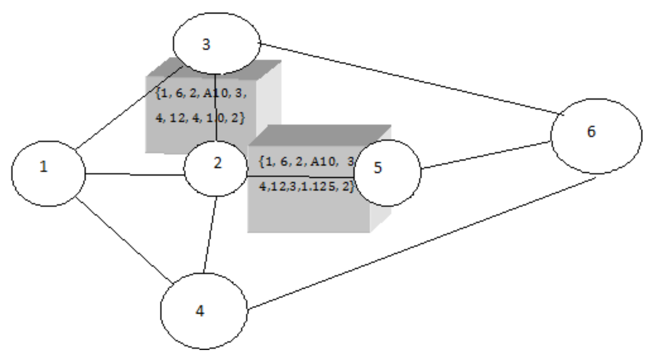

Figure 4. PROBE message sent by Node 2.

PROBE message for node 3-
{1, 6, 2, A10, 3, 4, 12, 4, 1.0, 2}
PROBE message for node 5-
{1, 6, 2, A10, 3, 4, 12, 3, 1.125, 2}
Node 3 has another message from 2

It makes the entries of probe message in its status table as Table 15.

Table 15. Status Table of Node 3

Predecessor	Cid	Path	Bandwidth	Delay	Hops
1	A10	1	4	.5	1
2	A10	12	4	1.0	2

Node 5 stores the information after receiving probe from node 2 as Table 16.

Table 16. Status Table of Node 5

Predecessor	Cid	Path	Bandwidth	Delay	Hops
2	A10	12	3	1.125	2

Node 4

Node 4 will not forward the message to 2 and 6, as the edges do not satisfy the bandwidth requirement.

Stage 3

Node 2

Node 2 will update the message received from 3 and forward it only to 5 as shown in Figure 5.

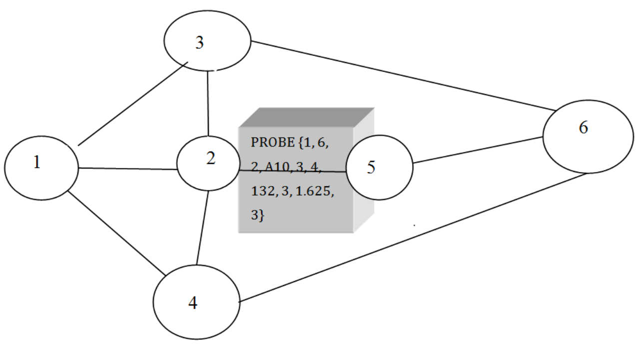

Figure 5. PROBE message sent by node 2.

The probe message that is sent by 2 is-
PROBE {1, 6, 2, A10, 3, 4, 132, 3, 1.625, 3}
Node 5 will receive another message from 2
It will update its status table as Table 17.

Table 17. Status Table of Node 5

Predecessor	Cid	Path	Bandwidth	Delay	Hops
2	A10	12	3	1.125	2
2	A10	132	3	1.625	3

Node 3

Node 3 will update & forward the message (received from 2) to 6 that have been represented in Figure 6. The probe message is.
PROBE {1, 6, 3, A10, 3, 4, 123, 4, 1.5, 3}
Node 6 will make entry into its status table as Table 18.

Table 18. Status Table of Node 6

Predecessor	Cid	Path	Bandwidth	Delay	Hops
3	A10	13	4	1.0	2
3	A10	123	4	1.5	3

Node 5

As Figure 7, Node 5 will forward first message received from 2 to 6 as-
PROBE {1, 6, 5, A10, 3, 4, 125, 3, 1.75, 3} .

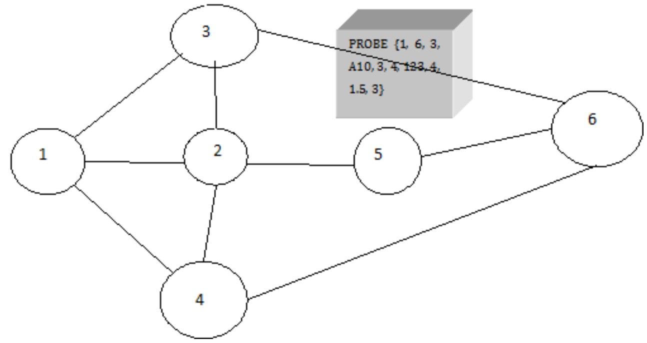

Figure 6. PROBE message sent by node 3.

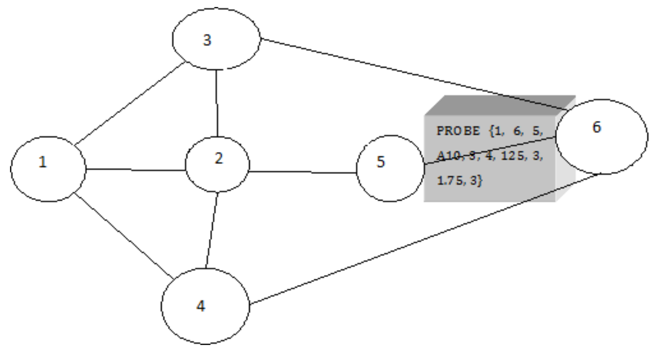

Figure 7. PROBE message sent by 5.

Node 6, destination has received three messages and its status table has been shown in Table 19.

Table 19. Status Table of Node 6

Predecessor	Cid	Path	Bandwidth	Delay	Hops
3	A10	13	4	1.0	2
3	A10	123	4	1.5	3
5	A10	125	3	1.75	3

Stage 4

Node 5

As shown in Figure 8, 5 will forward following message received from 2 to 6 -
PROBE {1, 6, 5, A10, 3, 4, 125, 3, 2.25, 4}

Thus, 6, destination has received four messages from 3, 5 and maintains all entries into status table that have been given in Table 20.
The paths found are-
1-3-6

1-2-3-6
1-2-5-6
1-3-2-5-6

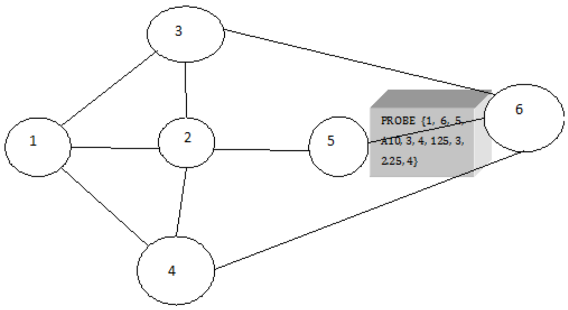

Figure 8. PROBE message sent by 5.

Table 20. Status Table of Node 6

Predecessor	Cid	Path	Bandwidth	Delay	Hops
3	A10	13	4	1.0	2
3	A10	123	4	1.5	3
5	A10	125	3	1.75	3
5	A10	1325	3	2.25	4

In the list of paths, now on destination, fuzzy logic has been applied to get one QoS metric from two metrics bandwidth and delay. If, the two paths will have same fuzzy value, the QoS path with fewer hops will be chosen.

3.2 Illustration of Finding Disjoint Paths Using Fuzzy Logic

3.2.1 Fuzzification of Inputs and Outputs

The two input variables to be fuzzified are bandwidth and delay. One output value i.e. QoS path significance value represented by FM (Fuzzy Metric) is produced from FLS when bandwidth and delay are given as input value. Each of the two input parameters for the algorithm, bandwidth and delay, are described using three linguistic variables: “Low”, “Medium” and “High”.The Table 21. shows the fuzzy sets for input variables.

Table 21. Fuzzy set of Input parameter

Input parameter	Fuzzy sets
Bandwidth	{Low, Medium, High}
Delay	{Low, Medium, High}

Here, as both the inputs have three linguistic terms correspondingly, hence a 3 by 3 rule matrix is prepared and represented for the output in the Table 22. Where, the columns representing “Low”, “Medium” and “High” bandwidth and the rows representing “Low”, ”Medium” and “high” Delay . Consequently, the antecedents of fuzzy IF-THEN rules can be originated by a 3 × 3 matrix.

Table 22. The Fuzzy Rule Matrix

			Bandwidth	
D		Low	Medium	High
e	Low	Medium FR1	Low FR4	Very low FR7
l	Medium	High FR2	Medium FR5	Low FR8
a	High	Very High FR3	High FR6	Medium FR9
y				

Thus, the output parameter, (FM) is described using five linguistic variables:

“Very Low”, “Low”, “Medium”, “High” and “Very High”. The FM has been represented in Table 23.

Table 23. Fuzzy set of Output parameter

Output parameter	Fuzzy sets
FM	{Very Low, Low, Medium, High and Very High}

Rule Structure:
The rules which describes the input output mapping are as follows-
RULES:
Rule 1: if Bandwidth is Low and Delay is Low then FM is Medium
Rule 2: if Bandwidth is Low and Delay is Medium then FM is High
Rule 3: if Bandwidth is Low and Delay is High then FM is Very High
Rule 4: if Bandwidth is Medium and Delay is Low then FM is Low
Rule 5: if Bandwidth is Medium and Delay is Medium then FM is Medium
Rule 6: if Bandwidth is Medium and Delay is High then FM is High
Rule 7: if Bandwidth is High and Delay is Low then FM is Very Low
Rule 8: if Bandwidth is High and Delay is Medium then FM is Low
Rule 9 if Bandwidth is High and Delay is High then FM is Medium

In the proposed fuzzy scheme, the path is considered as the best path having minimum value for FM. The procedure of rule structure has been formulated according to it.

The path having low delay and high bandwidth is measured as the best path. Thus, for the path with high bandwidth and low delay, the Fuzzy Metric value (FM) is very low (Rule-7) i.e. the best value and the corresponding path is the most significant path. In contrast, for the path with low bandwidth and high delay, the FM value is very high (Rule-3) i.e. the worst value and the path is the least significant path. The other rules are formed in the similar way.

3.2.2 Membership Functions

Each linguistic variable in the rule set of (low, medium, high) can be represented as a fuzzy set that is uniquely defined by a membership function. The membership function yields a degree of strength to each value belonging to a fuzzy set. The grade of membership is given between [0, 1]. The fuzzy rules use the input membership values as weighting factors to determine their influence on the fuzzy consequent represented by five linguistic variables. There are a number of shapes that can be used for the membership function of each input such as bell, triangular, trapezoidal and exponential shapes. We have chosen the triangular shape because of its simplicity.

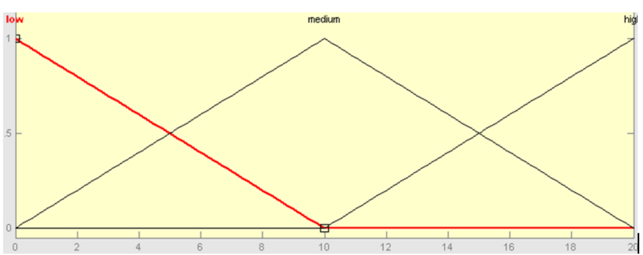

Figure 9. Membership function for bandwidth.

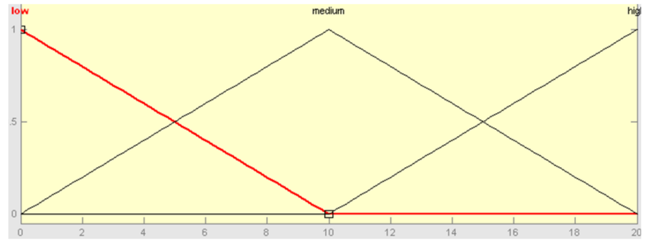

Figure 10. Membership function for Delay

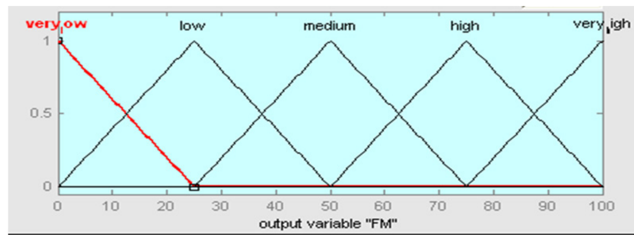

Figure 11. Membership function for FM.

The Figures 9-11 show the membership functions for the "bandwidth" (scale of 0–20), "delay" (scale of 0–20) and FM-output parameter (scale of 0-100).

The membership function for both input are defined as follows:

Here, the bandwidth is defined by the three parameters Low, Medium, High in range (0-20). Applying possibility distribution; the three parameters can have the following bounds-

Low=0
Medium=10
High=20

Similarly, for delay, the bounds are defined as follows-

Low=0
Medium=10
High=20

Using the above bounds, a mathematical function for the membership function of bandwidth and delay can be formulated as follows

$$F\,(\text{bandwidth}) = \begin{cases} 0 & : \text{bandwidth} < \text{low} \\ (\text{bandwidth} - \text{low}) / (\text{medium} - \text{low}) & : \text{low} <= \text{bandwidth} <= \text{medium} \\ (\text{high} - \text{bandwidth}) / (\text{high} - \text{medium}) & : \text{medium} <= \text{bandwidth} <= \text{high} \\ 0 & : \text{bandwidth} > \text{high} \end{cases}$$

$$F\,(\text{delay}) = \begin{cases} 0 & : \text{delay} < \text{low} \\ (\text{delay} - \text{low}) / (\text{medium} - \text{low}) & : \text{low} <= \text{delay} <= \text{medium} \\ (\text{high} - \text{delay}) / (\text{high} - \text{medium}) & : \text{medium} <= \text{delay} <= \text{high} \\ 0 & : \text{delay} > \text{high} \end{cases}$$

Using the defined fuzzy rules and the membership functions, we are calculating the degree of membership for the first path as below-

The bandwidth and delay values of first path are - 4 and 1

Thus, input values for the fuzzy system

Bandwidth=4

Delay=1

After applying the formula on the inputs,the obtained degree of member ship are

Input value 1(Bandwidth of first path) - 4

Degree of membership

Low=0.60, Medium=0.40,High=0

Input value 2(Delay of first path) - 1

Degree of Memebership

Low=0.90, Medium=0.10,High=0

Passing these values to fuzzy rule set ,we have triggerd all nine rules,as can be seen

Table 24. Fuzzy Result

Rule	Bandwidth	Delay	Fuzzy Resultant	Output Value
1	Low-0.60	Low-0.90	Medium	0.60
2	Low-0.60	Medium-0.10	High	0.10
3	Low-0.60	High-0	Very High	0
4	Medium-0.40	Low-0.90	Low	0.40
5	Medium-0.40	Medium-0.10	Medium	0.10
6	Medium-0.40	High-0	High	0
7	High-0	Low-0.90	Very Low	0
8	High-0	Medium-0.10	Low	0
9	High-0	High-0	Medium	0

in Table 24. Since each rule represents AND of two conditions,the output response values have been calculated as the minimum of memebership valuse of two places.

Those rules are considered as fired whose output value is Non-zero. Here four (1,2,4,5) out of nine rules have been fired. The output value has been taken as minimum value among bandwidth and delay.

These fuzzy output responses magnitudes are now composited and defuzzified to return actual crisp value. In the method that we have used for defuzzification, the weighted strengths of each output variables are multiplied by their respective output variable center points and summed. The area is then divided by the sum of the weighted member function strengths and the result was taken as the output.

The five output linguistic parameters have been calculated as:-

Very Low(VL)=max[Rule 7]=0

Low(L)-max[Rule 4,Rule 8]=max[0.40,0]=0.40

Medium(M)-max[Rule1,Rule 5,Rule 9]-max[0.60,0.10,0]=0.60

High(H)-max[Rule 2,Rule 6]=[0.10,0]=0.10

Very High(VH)-max[Rule 3]=[0]-0

The output center points of the defuzzification method have been taken as[10]

Very low center(Vlc)=0

Low center(Lc)=25

Medium Center(Mc)=50

High Center(Hc)=75

Very High Center(Vhc) =100

Hence the Fuzzy metric value for the first path is

Vlc * VL + Lc * L + Mc * M + Hc * H + Vhc * VH/ VL+L+M+H+VH

=0.40*25+0.60*50+0.10*75/(0.40+0.60+0.10)

=10+30+7.5/1.10

=47.5/1.1

=43.1

The fuzzy values for other paths have been calculated in the same manner.

The path table, thus obtained after implementing the fuzzy logic has been specified in Table 25.

Table 25. Path Table with Fuzzy value

Predecessor	Path	Bandwidth	Delay	FM	Hops
3	1-3-6	4	1.0	43.1	2
3	1-2-3-6	4	1.5	44.5	3
5	1-2-5-6	3	1.75	47.2	3
5	1-3-2-5-6	3	2.25	48.4	4

The path having the minimum FM value is the best path.

The process of making the disjoint table at destination can be described as follows:

Two lists need to be maintained to implement this process-path list and disjoint list.

Path list contains all the paths with their fuzzy value and disjoint list contains only disjoint paths with their fuzzy value.

- Sort the paths on the basis of fuzzy value.
- Take first path i.e. 1-3-6 and put it on to disjoint list.
- Take next path from path list and find whether this path has overlapping edges with the paths available in disjoint list. If so, do not put this path into disjoint list. Here the second path is 1-2-3-6. It has overlapping

edges with 1-3-6. So this path will not be included to disjoint list.

- Take next path 1-2-5-6. It has no overlapping edges with the path 1-3-6. So this path will be moved to disjoint list.
- Now the next path is 1-3-2-5-6. It has overlapping edges with the paths available in disjoint list. Hence, it will not be considered for disjoint list.

Therefore, the final disjoint path list with fuzzy values at destination will be

1-3-6 -- 43.1

1-2-5-6 -- 47.2

Now the destination will send the acknowledgement to source for all the disjoint paths with the fuzzy value. The source will send the packets on the found paths.

4. Conclusion and Future Work

In this paper, a distributed QoS multi-path routing algorithm (DQM) has been proposed which is based on DRA algorithm presented in[4]. DRA is a generic framework to find QoS path in a distributed way. In DRA, an intermediate node can forward at most one probe. After forwarding initial probe, it discards all the successively received probes. In the proposed DQM algorithm, all the probes are forwarded. As a result, at destination all QoS eligible paths from a source to destination are obtained. Amongst them, the most optimal QoS disjoint paths are selected using fuzzy logic. Here, the QoS criteria include – residual bandwidth and delay. These two QoS parameters (bandwidth and delay) have been converted into one QoS value using fuzzy logic. Based on this value, the list of calculated paths has been filtered to disjoint paths

This paper formulated a comprehensive distributed multiple disjoint QoS paths computational strategy. There are several research directions which require further investigations in the future. Some of them are pointed here.

4.1 Impact on Organization of the Network

In the proposed approach, we assumed that the topology information and current network state are known. Assumption has also been made about the domain of the network where the algorithm operates. As a result of this assumption, all the links from the source my lead to the destination .We plan to design more explorative mechanism to find the way out for this assumption. The solution remains for future work.

Another direction worthy of investigation is exploring the impact of the proposed strategy on the larger network since our study showed and illustrated network that is confined to small number of nodes.

4.2 Provision in Future Internet Protocol - IPv6

One future direction of the present research work is in the context of future Internet protocol –IPv6 that has incorporated Quality of Service.

IPv6 is an enhanced version of IPv4 protocol specification. The reasons for transition from IPV6 to IPv4 are - integrate QoS, security considerations, mobility issues and increased address usage in the network domain.

Thus, more investigations are required to study as to how the proposed approach results can be framed into message format of IPv6[11].

5. References

1. Upadhaya S, Devi G. Exploring issues for QoS based routing algorithms. International Journal on Computer Science and Engineering. 2010; 02(05):1792–5. ISSN: 0975-3397.
2. Upadhaya S, Devi G. Characterization of QoS based routing algorithms. International Journal of Computer Science and Emerging Technologies. 2010 Oct; 1(3):133–7.
3. Masip-Bruina X, Yannuzzib M, Domingo-Pascuala J, Fonteb A, Curadob M, Monteirob E, Kuipersc F, Van Mieghemc P, Avalloned S, Ventred G, Aranda-Gutierreze P, Hollickf M, Steinmetzf R, Iannoneg L, Salamatian K. Research challenges in QoS routing, Comuter Communications. 2006 Mar; 29(5):563–81.
4. Che S, Nahrstedt K. Distributed quality-of-service routing in high-speed networks based on selective probing, Tech. Report. Lowell MA: University of IL at Urbana-Champaign, Department of computer Science; 1998 Oct 11-14. p. 80–9.
5. Devi G, Upadhaya S. A comprehension study of distributed qos approach based on selective probing. International Journal of Advanced Research in Computer Science. 2014 May-Jun; 5(5):25–64.
6. Zhang R, Zhu X. Fuzzy routing in QoS networks, Fuzzy Systems and Knowledge Discovery. 2005; 3614:880–90.
7. Shehu A, Maraj A. Fuzzy logic approach for QoS routing analysis. Available from: http://www.intechopen.com/ 2012.
8. Upadhayay S, Sharma M. Realization of a new fuzzy mixed metric approach for optimizing routing decisions in packet switched networks. Journal of Theoretical and Applied Information Technology. 2008; 4(7):629–37.
9. Upadhayay S, Sharma M. Reinforcement of a new fuzzy mixed metric approach through fuzzy routing algorithms. International Journal of Computer Science and Network Security. 2008 Feb; 8(2):271–6

10. Cohen A, Korach E, Last M, Ohayon R. Fuzzy based paths ordering algorithm in networks with imperfect QoS information. IEEE. 2003; 2:1373–80.

11. IPv6 Addressing. 2008. Available from: http://www.cisco.com/web/strategy/docs/gov/IPv6_WP.pdf