World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:3, 2008

A Feasible Path Selection QoS Routing Algorithm with two Constraints in Packet Switched Networks

P.S.Prakash and S.Selvan

***Abstract*—**Over the past several years, there has been a considerable amount of research within the field of Quality of Service (QoS) support for distributed multimedia systems. One of the key issues in providing end-to-end QoS guarantees in packet networks is determining a feasible path that satisfies a number of QoS constraints. The problem of finding a feasible path is NP-Complete if number of constraints is more than two and cannot be exactly solved in polynomial time. We proposed Feasible Path Selection Algorithm (FPSA) that addresses issues with pertain to finding a feasible path subject to delay and cost constraints and it offers higher success rate in finding feasible paths.

***Keywords*—**feasible path, multiple constraints, path selection, QoS routing

I. INTRODUCTION

QoS based network architectures are being designed to provide QoS guarantees for various applications such as audio, video and data. IntServ (Integrated Services) is less scalable and robust than stateless network architecture Diffserv (Differentiated Services).IntServ provides end-to-end guaranteed or controlled load service on a per-flow basis, while DiffServ provides a coarser level of service differentiation among a small number of traffic classes. Many of these applications have multiple QoS requirements in terms of bandwidth, delay-jitter, loss, etc. Provision of QoS support at the network layer is by enabling QoS routing of data.

The problem of finding a path subject to two or more independent additive and/or multiplicative constraints in any possible combination is NP-Complete. The proof of NP-Completeness relies heavily on the correlation of the link weight metrics. QoS Routing is NP-Complete when the QoS metrics are independent, real numbers or unbounded integers.

In general, QoS routing focuses on how to find feasible and optimal paths that satisfy QoS requirements of various voice, video and data applications [9, 10].Hence, QoS routing is the first step that needs to be done in both reservation-based networks (e.g.,Intserv, MPLS) as well as reservation less ones(e.g.,Diffserv).In reservation-based networks, QoS routing is performed by source nodes to determine suitable paths for connection requests. In the case of Diffserv, the QoS –based routes can be requested by network administrators for traffic engineering purposes. In many cases, the problem of QoS routing is known to be NP-complete and thus mostly dealt with using heuristics and approximations. A survey on such solutions can be found in [11, 12].

A. *QoS Constraints*

The constraints can be classified into two types: Path constraints and Tree constraints [7]. The Path constraints need to be satisfied from the sender to the receiver. Tree constraints need to be satisfied over the entire multicast distribution tree created by the multicast routing protocol from the sender(s) to the receiver(s).

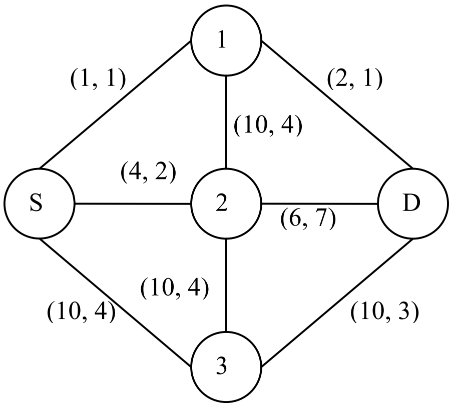

Fig 1 A Simple network topology

The computation complexity is primarily determined by the composition rules of the metrics [24]. The three basic composition rules are: additive (such as delay, delay jitter, logarithm of successful transmission, hop count and cost), multiplicative (like loss probability = probability of successful transmission) and concave/min-max (e.g., bandwidth).

The additive and multiplicative metric of a path is the sum and multiplication of the metric respectively for all the links

---

Manuscript received July 07,2008

P.S.Prakash is with Computer Science and Engineering (PG) Department, Sri Ramakrishna Engineering College, Coimbatore, TamilNadu, India.(Phone:+91 422 2312021, 99945 25625;e-mail: prakashpsrajan@rediffmail.com)

Dr. S. Selvan is with St.Peter’s Engineering College,Chennai, India.

constituting the path. The concave metric of a path is the maximum or the minimum of the metric over all the links in the path. In Fig 1, the '*S*' represents the source node and '*D'* represents the destination node. The link delay and bandwidth of each edge is shown in parenthesis.

B. Sources of Inaccuracy in Network State Information

QoS routing relies on state information specifying resource availability at network nodes and links, and uses it to find paths with enough free resources to accommodate new flows. In turn, the successful routing of new flows together with the termination of existing ones, induce constant changes in the amount of resources available [20, 21]. These must then be communicated back to QoS routing to ensure it makes its decision based on correct information. As a result, changes in resources availability are usually communicated either infrequently (e.g.,only when they are big enough) or imprecisely,(e.g., after aggregating network states). Such limitations can introduce substantial inaccuracy in the information used by path selection to identify good paths through the network, and, as stated before, we want to address such issues in this paper. However, before we proceed with our investigation, we briefly review some of the underlying parameters that determine the extent of inaccuracy that we can expect in network state information.

Limiting the number of entities (nodes and links) generating updates on their state, is a generic scalability issue that is not specific to QoS routing. Indeed, as network sizes grow, scalability quickly becomes a generic concern that has been the source of the many hierarchical schemes in use by network protocols.

The rest of the paper is divided as follows. Section II dealt with QoS Path Selection Problems, Section III handles network model and proposed algorithm. Section IV presents the simulation results obtained and comparative study with other algorithms and finally section V includes a brief conclusion.

II. QOS PATH SELECTION PROBLEMS

One of the key issues in providing end-to-end QoS guarantees in packet networks is how to determine a feasible path that satisfies a number of QoS constraints. The problem of finding a feasible path is NP-Complete [1] if number of constraints more than two and that cannot be exactly solved in polynomial time. Network layer has a critical role to play in the QoS provision process.

The approaches used by the QoS routing algorithms follow a trade-off between the optimality of the paths and the complexity of the algorithms especially in computing multi-constrained path [8].

A. Constraint Shortest Path (CSP):

The constraint-based path selection problem has been extensively researched in the last two decades, often in the computational theory and algorithms literature. The underlying problem can be stated as follows.

Consider a directed network G (V, E) where V is the set of nodes and E is the set of links of the network. Each link (u, v) ∈ E is associated with two integer weights $c_{uv} > 0$ (cost) and $d_{uv} > 0$ (delay).For any path p define cost as,

$$c(p) = \sum_{(u,v)\in p} c_{uv} \quad \text{and} \quad d(p) = \sum_{(u,v)\in p} d_{uv} \tag{1}$$

Given two nodes *s, d*, a path *p* is said to be feasible if $d(p) \le \Delta_d$ where Δ_d is delay bound between the source and destination nodes.

B. CSDP (k) (Constrained k shortest Disjoint-Paths)problem:

Here the objective is to find a set of k link-disjoint paths between a source 's' and a destination 'd' with minimum total cost and with the total delay satisfying certain pre-specified bound. This problem arises in the context of providing alternate QoS paths to achieve protection against link failures.

C. MP-DCP Problem (Most Probable Delay Constrained Path Selection under Inaccurate Information):

The Objective is to identify a path that has the highest probability of satisfying a delay bound. The delay of each link is a random variable. This problem is of great importance since accurate state of a network (parameter information) is not often available.

The CSP problem is NP-complete [2]. So many heuristic approaches and approximate algorithms have been proposed. In general, Heuristics do not provide performance guarantees on the quality of solution produced-approximation algorithms deliver solution within arbitrarily specified precision requirement.

A widely studied case of the Restricted Shortest Path (RSP) problem group is the Delay-Constrained Least Cost (DCLC) problem. The DCLC problem is NP-complete.

In [4], a source routing algorithm called Dual Extended Bellman-Ford (DEBF) to solve the DCLC problem is proposed. The heuristic used by the DEBF algorithm is a bi-directional application of the Extended Bellman-Ford (EBF) algorithm. In the forward phase the EBF is applied to compute the least cost feasible path from the source to the destination using the delay metric. If a path that satisfies the delay constraint is found on this phase, the cost metric is used in the backward phase. The DEBF algorithm executes the EBF algorithm twice, and so it has a complexity of O(2nm)

D. CSP Algorithms

1) Delay Constrained Unicast Routing (DCUR) algorithm

The DCUR algorithm proposed by *Salama* and *Reeves* [3] uses a heuristic to compute delay constrained least cost paths in a distributed manner. The DCUR algorithm is based on the Distance-Vector algorithm and it maintains a cost and delay vector at every node by a distance vector protocol. A delay constrained path is constructed using a control message sent from the source node to the destination node [23]. The algorithm has a complexity of $O(n^3)$. Improvements on algorithm proposed in [3] are by avoiding loops instead of detecting and removing loops.

2) ε - Optimal approximation algorithm

To deal with NP-complete problem the approach is to look for a polynomial–time algorithm that guarantees finding an approximate solution to the optimal one. An algorithm is ε - Optimal if it returns a path whose cost is at most (1+ε) times the cost of the optimal path. Approximation algorithms perform better in minimizing the cost of a returned feasible path as ε = 0. Smaller values of ε lead to an increased complexity.

3) Lagrangian–based Linear Composition Algorithm

The algorithm used in the Lagrangian - based linear composition approach [22], linearly combines the delay and cost of each link and finds the shortest path with respect to the composite measure. Thus, the weight of a link becomes

$$w(u,v) = \alpha.d(u,v)+\beta.l(u,v) \qquad (2)$$

where, α and β are called the multipliers. A key issue in the Lagrangian–based composition approach is how to find appropriate values for the multipliers.

4) SSR+DCCR Algorithm

In [5], A heuristic is proposed in which the cost for the least –delay path is selected as the cost constraint. The problem is then solved by minimizing a nonlinear length function that gives more priority to lower-cost paths. DCCR algorithm was proposed to minimize this nonlinear length function. DCCR is a *k* shortest path-based algorithm. The performance of the DCCR algorithm depends on the *k*. The performance of algorithm is good for larger values of k[6].The SSR+DCCR algorithm, reduces the search space and tighten the cost bound by using a Lagrangian-based algorithm before applying DCCR.

III. PROPOSED WORK

A. Network Model

Consider a network that is represented by a directed graph, G= (V, E), where V is a set of vertices and E is a set of edges. Each link (u, v) is represented by two non-negative measures namely a delay δ (u, v) and a residual bandwidth *R* (u, v). Given a delay bound *d*, a delay-jitter bound *j*, and buffer space bound b_m for all *m* nodes in the network, a leaky bucket <σ,b>, where σ is the average token rate and b is the maximal burst rate in bytes (token bucket size) and a bandwidth to reserve *r* (r ≥ σ) to reserve, find a path *p* such that,

$$D(p, r, b) \leq d,\ J(p, r, b) \leq j,\ B(p, r, h) \leq b_h \qquad (3)$$

for all nodes along the path, where b_h is the buffer space constraint for the node with *h* hops from the source and $r \leq R_j$ ($\forall$ R_j ε p) where R_j is the link residual bandwidth. End-to-end delay, delay-jitter and buffer space constraints are determined by bandwidth allocation when rate proportional service disciplines such as Weighted Fair Queuing are used. Finding a path that satisfies end-to-end delay, delay-jitter and buffer space constraints is solvable in polynomial time when the relationship between constraints in taken into account. Bandwidth being reserved and burst ness of the traffic source is the two factors that determine the end-to-end queuing delay of the flow. For the above mentioned rate proportional service disciplines, if the traffic source is constrained by a leaky bucket < σ, b >, where σ is the average token rate and b is the maximal burst rate in bytes (token bucket size) then for a path p with *n* hops and link capacity C_i at hop i, the *provable end-to-end delay* as given by Zhang [19] is:

$$d(p,r,b) = \frac{b}{r} + \frac{h * L_{\max}}{r} + \sum_{i=1}^{n} \frac{L_{\max}}{C_i} + \sum_{i=1}^{n} prop_i \qquad (4)$$

Where $L_{\max}$ = Maximum packet size in the network
$prop_i$ = Propagation delay of link *i.*
(r ≥ σ) = Bandwidth reserved.
b = burst rate in bytes
h = hop count

Here we use additive distance function by using the link cost *l*(i) and length function *l*(p) where,

$$l(i) = \frac{L_{\max}}{r} + \frac{L_{\max}}{C_i} + prop_i \qquad (5)$$

$$l(p) = \frac{b}{r} + \sum_{i \in p} l(i) \qquad (6)$$

The goal now is to find a path *p,* such that $l(p) < \Delta_d$ using the proposed algorithm . If $l(p) > \Delta_d$ then there is no path that meets the given delay bound Δ_d. A path *p* satisfying the delay, delay-jitter and buffer space requirement is said to exist, if and only if the length of path p as defined by (4) is less than or equal to delay bound *d* (i.e., $l(p) < \Delta_d$) and each node along the path 'p' meet the delay bound as defined by (3). If such a path exists then the path *p* is said to be a feasible path.

B. Selecting Feasible Path

Our Feasible Path Selection Algorithm (FPSA) finds a feasible path if one such exists. However, there might be more than one feasible path available in the network. In order to efficiently utilize the network resources we should select the feasible path, which consumes less network resources among the available multiple feasible paths. Therefore, we need some optimality criteria in choosing a feasible path among multiple feasible paths. The optimality criteria that can be considered are minimum hop count, least delay, throughput, and bandwidth. A feasible path can be selected using one of the four optimality criteria mentioned above independently or using a combination of them with priorities. A few combinations of optimality criteria that can be used to achieve efficient utilization of network resources are *shortest-delay path, widest-shortest path, shortest-widest path and shortest-minimum-bandwidth path* [18].

Here in our work we considered the bandwidth and delay as prime constraints to choose the feasible paths. This work mainly focuses on the Diff Serv and its policing. DiffServ is proposed to provide QoS on the Internet, while solve the scalability problem with IntServ. In DiffServ framework, the routers supporting DiffServ form a DiffServ domain.

C. Algorithm Description

Every node acquires the underlying network G(N, L), where each link *l* ε L, and N being the number of nodes.

```
FPSA (G ,s , d, c ( bw, $\Delta_d$))
Step1: Begin :
    G: network
    s : source
    d : destination
    m : metrics
Step2: Qos_P ($\Delta_d$, s , d )
Step 3: Enable source routing
(Let m ($n_1$ , $n_2$ ) be a metric for link($n_1$ ,$n_2$ ).
For any path P = ($n_1$ , $n_2$ , ..., $n_i$ , $n_j$ ),
 metric m is:
Step 4:
a) additive, if m(P) = m($n_1$,$n_2$)+m($n_2$,$n_3$ )+...+m($n_i$ ,$n_j$)
b) multiplicative, if m(P) = m($n_1$,$n_2$)*m($n_2$,$n_3$)*...*m($n_i$, $n_j$)
c) concave, if m(P) = min{m($n_1$ ,$n_2$),m($n_2$ ,$n_3$ ), ... ,
m($n_i$ ,$n_j$ )}
Step 5: Qos_Routing P(bw, s , d)
Step 6: get-bw(s,d)
available bw = total bandwith * utilization - reserved
bandwith;
route bandwidth = utilization- available bandwidth
return bw,m
Step 7: constrained route C (bw ,s ,d)
if (success =1) then, // Feasible path found
return F_path
else
set success=0 // No Feasible path found
return no F_path
 End
```

Fig. 2 Pseudo code of FPSA

The proposed algorithm first finds the feasible path specified by the two constraints namely delay and bandwidth. A feasible path is found and then the data are sent along that feasible path. Fig.2 shows the pseudo code of the proposed algorithm.

In step 1, the source node *s*, destination node *d,* the additive metric delay and concave metric bandwidth are first initialized. In step 2, the initialized parameters are considered for finding the feasible path satisfying the Quality of Service (QoS) requirements.

The main idea here is to identify the feasible path that satisfies both the metrics (i.e, delay and bandwidth). The link cost and the path length function are calculated based on equations (4) and (5) mentioned above. Where L_{max} represents Maximum packet size in the network, $prop_i$ represents Propagation delay of link *i*, the factor $r \geq \sigma$ identifies the Bandwidth reserved, b represents burst rate in bytes, and 'h' being the hop count. The path having larger bandwidth and less delay is taken for routing the packets in a network. The success rate (SR) in finding the feasible path is given by

$$\text{SuccessRate(SR)} = \frac{\textit{Number of packets successfully received}}{\textit{Total number of packets sent}} \qquad (7)$$

The success rate in finding the feasible path in the proposed algorithm shows excellent results. For the increase in packet size the success ratio also increases, when compared with the other two algorithms.

To find the feasible path in a network we calculate the bandwidth using,
*available bandwidth = total bandwidth * utilization - reserved bandwidth.* (8)

The path with sufficient bandwidth and less delay is returned as the feasible path. If a feasible path is found then the algorithm returns that path and the success is 1. If the algorithm is not able to find the feasible path then the algorithm terminates giving success as zero and no path found in the network.

IV. SIMULATION RESULTS

In the simulation model, a network topology is constructed with sufficient source and destination nodes. The constraints such as bandwidth and delay, connection request between source and destination are generated. The success rate (SR) is used as one of the prime criteria to determine the performance of various path selection algorithms. The algorithm was simulated in a discrete event C++ simulator. The success rate was measured as a ratio of packets received with packets sent. In this paper, we compare our algorithm with other algorithms namely Lagarangian Linear Composition Algorithm (LLCA) [13, 14, and 15] and Hybrid Multi Constraints Optimal Path Algorithm (HMCOP) [16, 17].

Fig.3 shows the topology model used, where edge routers (e_1 and e_2) shown as hexagon, core routers (c_1 and c_2) are square shaped. Source and destination nodes are arranged left and right respectively.

TABLE I
SUCCESS RATE IN FINDING FEASIBLE PATH

Number of Packets (x10^3)	Success Rate (%)		
	HMCOP	FPSA	LLCA
5	91.57	94.41	84.01
10	92.39	94.74	84.57
15	93.07	95.15	86.12
20	93.66	95.82	87.0
25	94.17	96.51	89.1
30	94.62	96.53	91.2
40	94.96	96.6	91.4
50	95.96	96.86	92.5

A. Performance Comparison based on Success Rate

Fig.4. shows the Success Rate of three algorithms. When the number of packets increases it is observed that the success rate for the proposed algorithm also improved. When the routing information is more accurate, the success rate in finding the feasible path also increases.

We varied the number of packets sent by a source to a destination and with the increase in number of packets, the Success Rate in finding the feasible path from ‘s’ to ‘d’ in our algorithm (FPSA) crosses above 95%, for number of packets being sent above 15000, showing the better performance compared to the HMCOP and LLCA.

Table I. represents the values of success rate of the proposed algorithm and the other two compared algorithms.

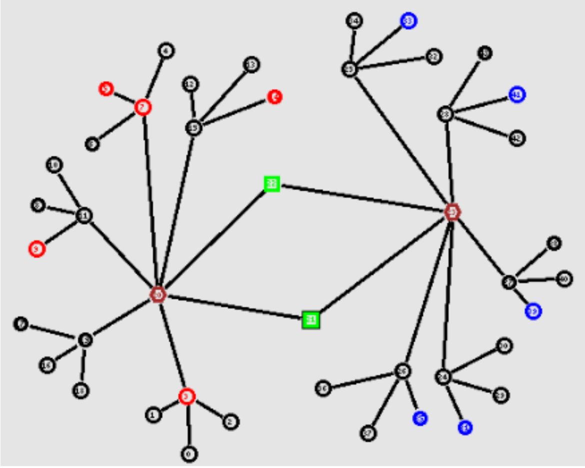

Fig. 3 The Topology Model

The packet drops in the FPSA algorithm is very less since the bandwidth used by this algorithm on the links is sufficient enough to transmit all packets to its destination without any loss in packet. This also shows in Fig.7 through Fig.9 that the various classes of traffic can be scheduled at different time intervals. The number of packets successfully received by the destination using HMCOP and LLCA algorithm is comparatively less, showing more drops in packets. From the simulation, it was studied that the bandwidth utilized by the other two algorithms are not sufficient enough to carry inelastic data packets since these algorithms are not allowing scaling these traffic flows. This shows that, as the number of packets being sent along the links increases, the drop in the packets also increases hence, lowering the success rate. The success rate for number of packets sent is at least 5,000 the success rate is least (84%) in LLCA. The number of packets sent for all the algorithms are the same. The delay in finding the feasible path should be less for any algorithm to prove its efficiency in finding a feasible path in a given network. Here in our simulation, we used four source nodes and four destination nodes. We calculated the end-to-end delay ‘*d*’ from (4), where L_{max}, the maximum packet size in the network ranges from 5,000 to 50,000. In our simulation, the bandwidth to be reserved for a link is considered to be 5MB.

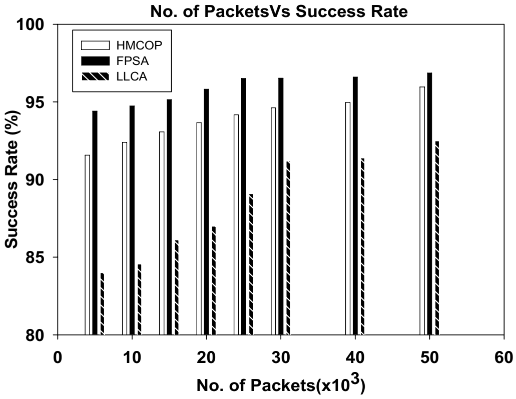

Fig. 4 Performance Comparison based on Success Rate

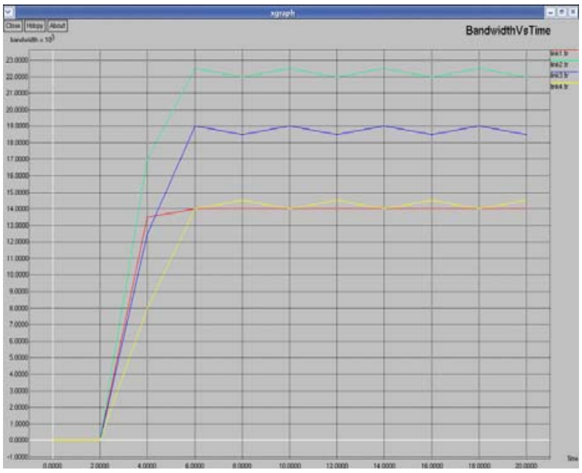

Fig. 5 Time Vs Bandwidth in FPSA

The network topology, considered here has a minimum of three routers and hop count is at least three. The propagation delay in each link is taken separately for calculation purpose. To find the feasible path in a network the delay calculation shows that our algorithm took nearly 40ms for numbers of packets sent being 50,000, while it took 71.4ms for the HMCOP and 73ms for LLCA algorithm.

The link cost $l(i)$, is found for finding the path length $l(p)$. Only for a path length $l(p) < \Delta_d$, the delay bound , the path is said to have satisfied the first constraint. The bandwidth on that link is calculated to find if its exactly a feasible path or not. The bandwidth can be found using (7). The burst rate used here is at least 500 bytes. All the algorithms were simulated individually while the number of packets being sent is varied for each simulation. Fig.6 shows the variation in delay with time for the three algorithms. In Fig.7. the proposed algorithm has throughput value of nearly 8.4 as the time progresses.

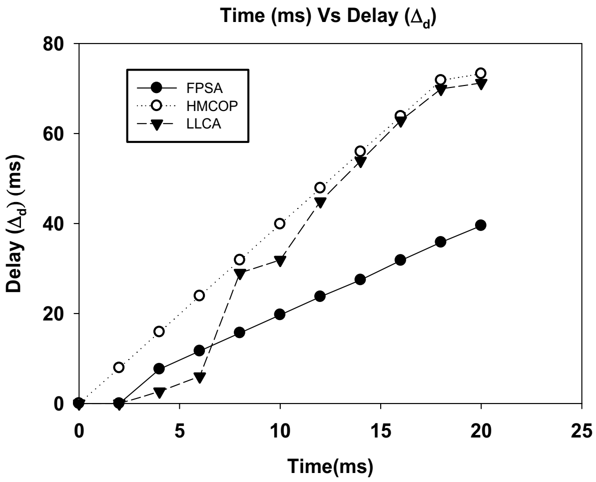

Fig. 6 Graph showing variation in Delay Vs Time

The throughput value of HMCOP algorithm has got only very slight variation. Compared with HMCOP the LLCA algorithm has got 65% of higher throughput, since the packet drops in the HMCOP is more than the other algorithm.

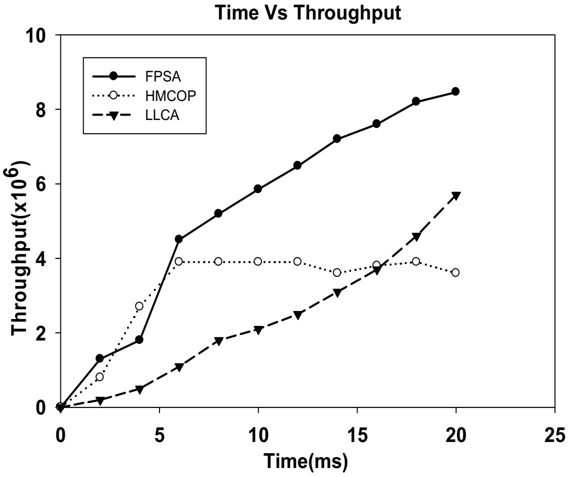

Fig. 7 Graph showing Time Vs Throughput

B. Performance Comparison based on Bandwidth

In Fig.8.and Fig.10. the link 2 for both LLCA and HMCOP algorithms are having the normal flow meaning that link2 is having fewer drops in the number of packets being sent and hence the bandwidth used by this link is nearly 40MB. The rest of the links shows slight variation in the bandwidth as the simulation time periodically increases because the links are with aggressive flows. So more packets have to be queued and less priority packets have to be dropped. This shows that both HMCOP and LLCA fail to meet the QoS requirements for inelastic applications.

The propagation delay for each link ranges from 1ms to nearly 10ms. In Fig.9 our algorithm is having moderate bandwidth in almost all the links, since all the packets being sent are received by the destination without many packets being lost.

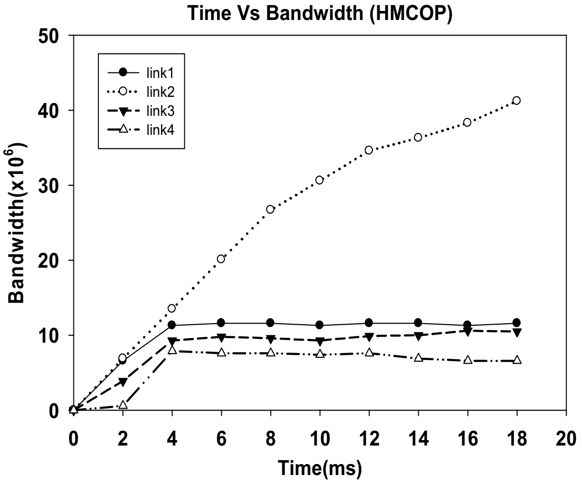

Fig. 8 The available bandwidth for each link in HMCOP

Moreover the amount of bandwidth in each link is above 10MB. This shows that all real time traffic can be accommodated in different links with negligible packet loss.

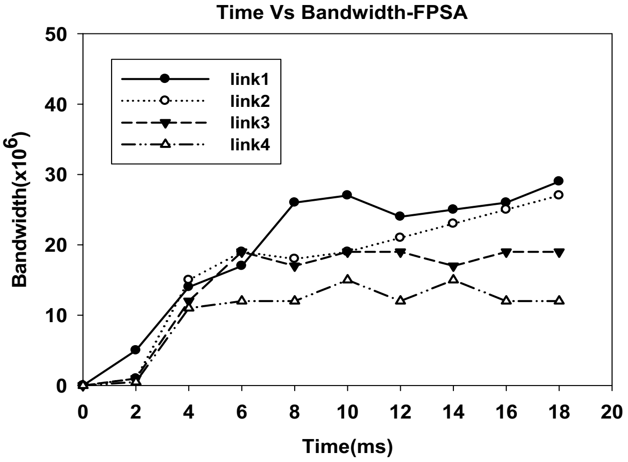

Fig. 9 Graph showing Bandwidth variation for FPSA at various time intervals

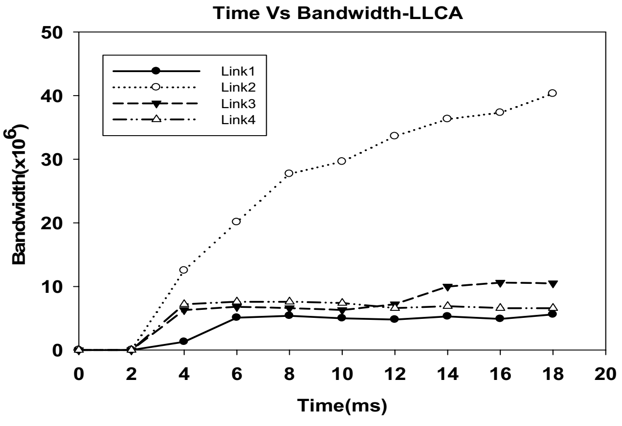

Fig.10 Graph showing Bandwidth variation for LLCA at various time intervals

V. CONCLUSION

We studied the problem of finding a feasible path subject to two constraints and our algorithm is supported by performance bounds that reflect the effectiveness of the algorithm in finding a feasible path. We analyzed the performance of the algorithm through simulations by calculating the success rate (SR) taking number of packets into consideration. When number of packets is increased our FPSA algorithm has still high success rate than other algorithms with the same number of packets. Also when compared with other algorithms such as LLCA and HMCOP, the proposed algorithm has obtained less delay and bandwidth in finding the feasible path for routing of packets.

REFERENCES

[1] Turgay Korkmaz, Marwan Krunz, Spyros Tragoudas, "Efficient Algorithm for Finding a Path Subject to Two Additive Constraints"., In Computer Communication Journal Vol 25, No. 3 pp. 225-238, Feb 2002.

[2] Z. Wang and J. Crowcroft, "Quality-of-Service routing for supporting multimedia applications," *IEEE JSACs*, vol.14, no.7, pp.1228-1234, Sept. 1996

[3] H.Salama, D. Reeves, Y.Viniotis, "A distributed Algorithm for Delay-Constrained Unicast Routing", INFOCOM'97, Japan, March 1997.

[4] G. Cheng and K.Ansari, "A New Heuristics For Finding The Delay Constraints Least Cost Path ", Proceedings of IEEE GLOBECOM 2003, San Francisco, CA,USA, Dec.2003, pp. 3205-3210.

[5] L.Guo, I.Matta, "Search space reduction in QoS routing", Proceeding of the 19th IEEE International Conference on Distributed Computing Systems, IEEE,1999 pp.142-149.

[6] F.A Kuipers, Korkmaz, M.Krunz, P.Vam Mieghem, "Overview of Constriant-Based Path Selection Algorithm for QoS Routing", IEEE communication magazine, Dec 2002.

[7] Cristina Aurrercoechea, Andrew T.Campbell and Linda Hauw, "A survey on QoS Architectures" ,IEEE/ACM Multimedia Systems, pp 138-151, May 1998.

[8] M. Curado, E. Monteiro, F. Kuipers, P. Van Mieghem, et.al., "Research challenges in QoS routing" Computer Communications, Volume 29, Issue 5, Pages 563-581, Mar 2006.

[9] D. Ghosh, V. Sarangan, R. Acharya, "Quality-of-service routing in IP networks", IEEE Transactions on Multimedia 3 (2) (2001) 200–208.

[10] G. Huston. "Next steps for the IP QoS architecture" Technical Report RFC 2990, IETF, November 2000.

[11] P. Van Mieghem, F.A. Kuipers, "Hop-by-hop quality of service routing", Computer Networks 37 (3–4) (2001) 407–423.

[12] F.A. Kuipers, T. Korkmaz, M. Krunz, P. Van Mieghem, "Performance evaluation of constraint-based path selection algorithms", IEEE Network 18 (5) (2004) 16–23

[13] A.Juttner, B. Szviatovszki, I. Mecs, Z. Rajko, "Lagrange relaxation based method for the QoS routing problem", Proceedings of the INFOCOM 2001 Conference, vol. 2, IEEE, 2001, pp. 859–868.

[14] G.Y. Handler, I. Zang, "A dual algorithm for the constrained shortest path problem" Networks 10 (1980) 293–310.

[15] M. Mehlhorn, K. Ziegelmann, "Resource constrained shortest path". In Proceedings of 8th European Symposium on Algorithms (ESA2000), 2000.

[16] T. Korkmaz, M. Krunz, "Multi-constrained optimal path selection",Proceedings of the INFOCOM 2001 Conference, vol. 2, IEEE, Anchorage, Alaska, 2001, pp. 834–843.

[17] T. Korkmaz, M. Krunz, "Routing multimedia traffic with QoS guarantees", IEEE Transactions on Multimedia 5 (3) (2003) ,429–443.

[18] S. Chen and K. Nahrstedt, "Distributed QoS routing with imprecise state information". ICCCN 1998, pp. 614–623.

[19] P. Van Mieghem (ed.), F.A. Kuipers, T. Korkmaz, M. Krunz, M. Curado, E. Monteiro, X. Masip-Bruin, J. Solé-Pareta, and S. Sánchez-López. "Quality of Service Routing", Chapter 3 in *Quality of Future Internet Services, EU-COST 263Final Report*, edited by Smirnov et al. in Springer LNCS 2856, pp. 80-117, 2003.

[20] R.Guerin, A.Orda, "QoS routing in networks with inaccurate information:theory and algorithms", IEEE/ ACM Transactions on Networking 7(3) (1999) 350-364.

[21] Turgay Korkmaz, Marwan Krunz, " A randomized algorithm for finding a path subject to multiple QoS requirements" , In computer networks Journal 36 (2001) pp 251-268.

[22] Orda, A. Sprintson,"Precomputation schemes for QoS routing", IEEE/ACM Transactions on Networking 11 (4) (2003) 578–591

[23] Krishnaiyan Thulasiramn,Ying Xiao, Guoliang Xue, "Advances in QoS Path(s) Selection Problem", in processing of IEEE INFOCOM 2005 Conference,pp.164-167.

[24] Yanxing Zheng, Turgay korkmaz, W.Dou," Two additive-constrained path selection in the presence of inaccurate state information", In Computer Communications Journal, Vol. 30, pp. 2096-2112, May 2007.

P.S.Prakash received B.E (EEE) and M.E (Electrical Machines) degrees from P.S.G College of Technology, Coimbatore, India and also received M.E degree in Computer Science and Engineering from Bharathiar University, India. His research interests include QoS Scheduling, Routing, Network Security and Management. At present, serving as Asst.Prof at Sri Ramakrishna Engg. College, Coimbatore, India.

Dr. S.Selvan is serving as Principal at St.Peter's Engineering College, Chennai, India. He has 30 years of teaching experience. He has published more than 75 papers in international and national journals and conference proceedings. His areas of research include, computer networking digital signal processing.