Asian Journal of Information Technology 15 (19): 3810-3819, 2016
ISSN: 1682-3915


# Optimized Multi Constrained Path Computation for QoS Routing

P.S. Prakash
Department of Computer Science and Engineering, RVS Technical Campus, Coimbatore, Tamil Nadu, India

**Abstract:** The goal of Quality of Service (QoS) routing algorithms is to find a feasible path satisfying a given set of constraints on parameters like delay, jitter and reliability and so on and one of the key issues in providing end-to-end QoS guarantees in packet networks is determining feasible path that satisfies a number of QoS constraints. We present our algorithm Optimized Multi-Constrained Routing (OMCR) for the computation of constrained paths for QoS routing in packet switched networks. OMCR applies distance vector to construct a shortest path for each destination with reference to a given optimization metric from which a set of feasible paths are derived at each node. OMCR is able to find feasible paths as well as optimize the utilization of network resources. OMCR operates with the hop-by-hop, connectionless routing model in Internet Protocol and does not create any loops while finding the feasible paths.

**Key words:** QoS routing, optimization, feasible path, multiple constraints, OMCR

## INTRODUCTION

Real Time Interactive applications such as video conferencing, video streaming and VoIP etc. require packets to be delivered to the destination within the stipulated time frame. The problem of finding a path subject to two or more independent additive and/or multiplicative constraints in any possible combination is NP complete. In general, QoS routing focuses on how to find feasible and optimal paths that satisfy QoS requirements of various voice, video and data applications (Ghosh *et al.*, 2001). The objective of QoS routing is to find feasible paths from source to destination that satisfy multiple constraints such as end-to-end delay, jitter, bandwidth, reliability and routing the traffic in such a way that network resources are utilized effectively (Prakash and Selvan, 2008). This is known as multi-constrained path optimization (Wang and Crowcroft, 1996). QoS Routing approaches have many bottlenecks such as update of global network state at appropriate time at each node. Some approaches proposed earlier have assumed that the dissemination of routing constraints is known (Mieghem *et al.*, 2001) but it is difficult to achieve in practice. This is due to dynamic nature of network and parameters such as residual bandwidth, queue lengths and propagation delay, addressing QoS routing subject to only a single constraint such as delay or bandwidth etc. computing only shortest paths without satisfying the multiple constraints simultaneously (Wang and Nahrstedt, 2002) and considering either multi-constrained path computation or optimization even though they are very much related to each other. Hence, it is important that a solution is required for distributed multi constrained routing that does not require global network state to be made available at each and every node and finds multi-constrained paths while optimizing the overall routing performance according to the given optimization metric. The path measurement of additive and multiplicative metric equals to the sum or product of its values of links along the path (Smith and Aceves, 2004; Yuan, 2002). Multiplicative metrics can be converted into additive metrics. The link metric can be chosen to be negative of the logarithm of the link availability, where the logarithmic function is used in general, to convert a multiplicative metric to an additive metric (Sobrinho, 2001). The path measurement of a minimal metric link bandwidth is determined by the minimal value of this metric of all links in the path.

Mieghem and Kuipers (2004) and Smith and Aceves (2004) has shown that maintaining only non-dominated paths for each destination in sufficient to compute constrained feasible paths.

**Literature review:** Chen and Nahrstedt (1998a, b) proposed to scale one component of the link weight down to an integer that is <$[W_i.x/C_i]$ where ‘x’ is a pre-defined integer and ‘$C_i$’ is the corresponding constraint on weight component $W_i$. They prove that the problem after weight scaling is polynomially solvable by an extended version of Dijkstra’s or Bellman-ford algorithm. Path selection subject to two constraints is not strongNP-complete and the computation complexity depends on the values of link weight in addition to the network size. Jaffe (1984) was the first to use a linear link-cost function $W(u, v) = \alpha W_1(u, v)+\beta W_2(u, v)$, in which $\alpha\beta \in Z^+$. The major limitation of this approach is that the ability to find feasible paths based on an aggregated metric largely depends on the quality of the function being used and most of them are empirical heuristics. Consequently, the shortest path computed with regard to a singular metric may not simultaneously satisfy the multiple constraints being considered. Based on TAMCRA (Neve and Mieghem, 2000) and its exact modification SAMCRA, (Mieghem *et al*., 2001) proposed a hop-by-hop destination based QoS routing protocol which has a worst-case complexity $O(kNlog(kN)+k^2CE)$, where 'k' is the number of non-dominated paths maintained at each node and C is the number of constraints being considered. A node that runs this protocol uses a modified Dijkstra's algorithm to compute 'k' non-dominated paths for each destination based on a nonlinear weight function $W_p = \max [W_i(p)/C_i]$, where $C_i$ the constraint on metric is $W_i$. This algorithm requires global network state at each node and routing constraints must also be known a priori.

Sobrinho (2001) adopts an algebraic approach and investigates the path optimization problem in the context of hop-by-hop QoS routing where routing is separated into path weight functions that define routing optimization requirements and the algorithms that compute the optimal paths based on the aggregated metric defined by the weight function being used (Sobrinho, 2002) establishes the algebraic properties that a path weight function must have for any routing algorithm to converge correctly. Smith and Aceves (2004) proposed an improved algebra to address multi-constrained path computation and the proposed generalized Dijkstra's algorithm has a complexity of $O(NW^2A^2)$ where W is the maximal value of link weights and A is the maximal number of true assignments in the traffic algebra. In Yuan (2002) shown that $O(N^2log(N))$ paths need to be maintained to have high probability of finding feasible paths.

We present Optimized Multi-Constrained Routing (OMCR) Algorithm which finds paths without loops subject to multiple constraints and nodes need not to maintain the global network state. This may be accomplished by ordering of routes based on shortest distances which are optimization metrics obtained from the various weight components of links like delay, reliability and bandwidth etc. and tracking the weight of every path reported by neighboring nodes when establishing shortest paths to derive a set of non-dominated paths for each destination.

## MATERIALS AND METHODS

### OMCR

**Network model:** A network is modeled by a connected directed graph $G = \{V, E\}$ , where V is the set of nodes and E is the set of edges interconnecting the nodes. We assume that each link $l_{u,v}$ is associated with a link weight vector $W(u, v) = \{W_1W_2,..., W_k\}$ in which $W_1$ is an individual weight component like single routing metric. Accordingly, any path 'p' from a source 's' to a destination 'd' can be assigned a path weight vector $w_p=\{w_1^p,w_2^p,...,w_k^p\}$ where $W_i^p = \sum_{lu,\ vW}(u,\ v)$, if $W_i$ is an additive metric such as delay; $w_i^p = \min(W_i(u,v)), l_{u,v} \in p$ or P if $W_i$ is a minimal metric such as bandwidth. The shortest distance of path 'p' is given by a function f(p) based on the weights of its links and f(p) is usually used to specify how the routing should be optimized. We denote f(p) as optimization function and shortest distance computed by f(p), the optimization metric given that the path with shortest feasible path is also optimal with regard to the optimization requirements given by f(p) and optimization metric and routing metric are not necessarily the same.

**Algorithm:** Let $D_{i,j}$ denote the distance between nodes 'i' and 'j' as known by the node 'i'. $D_{jk}^i$ denotes the shortest distance $D_j^k$ from node 'k' which is a neighbor of node 'i', to destination 'j' as reported to node 'i' by node 'k'. $SFD_{ij}$ denotes shortest feasible distance of node 'i' for destination 'j' which is an estimate of minimum shortest distance maintained for destination 'j' by node 'i'. A node 'i' maintaining a routing entry for each destination 'j' which includes $SFD_{ij}$, $D_{ij}$ and the successor set chosen for 'j' and denoted by $S_{ij}$. Node 'i' maintains a neighbor table that records the shortest distance $D_{jk}^i$ reported by each node 'k' in its neighbor set $N^i$ for each destination 'j'; and a link table that reflects the link state W(i,k) for each adjacent link $l_{i,k}$, $k \in N^i$. Each node must run OMCR for each destination we focus on the operation of node i's computation of the set of feasible paths for destination 'j'. Each node maintains 'x' feasible path for destination 'j', node 'i' may receive and record 'x' values of ?from each neighbor 'k'; node 'i' also reports to its neighbors the shortest distance of 'x' feasible paths from itself to destination 'j', of which the minimum value is also used as the shortest feasible distance $SFD_{ij}$ of node 'i'. For destination 'j'we have $D_{jj} = 0$, $SFD_{jj} = 0$ and $D_{jj}^k$ and when a node is activated, SFD is set to infinity which is defined by f(p) and all the other entries are set to empty. When node 'i' receives $D_{kj}$ from neighbor 'k', either updates the estimates $D_{jk}^i$ and will not disturb other estimates or node 'i' updates $S_{ij}(t)$ and $SFD_{ij}(t)$ for destination 'j' based on the equation:

$$S_{ij}(t) = \left\{k \middle| D_{jk}^i(t) \prec SFD_{ij}(t), k \in N^i\right\} \quad (1)$$

and updates its shortest feasible distances as follows

$$SFD_{ij}(t) = \min\left(D_{jk}^i(t) \| sd(i,k)(t)\right) \quad (2)$$

for all $D_{kj}$ reported by each neighbor 'k' and overall neighbors in $N^i$ or node 'i' remains idle. 'sd' is shortest distance of the adjacent link $l_{i,k}$. The two links are combined by the optimization function and compute the shortest distance of the resulting combined path. Apart

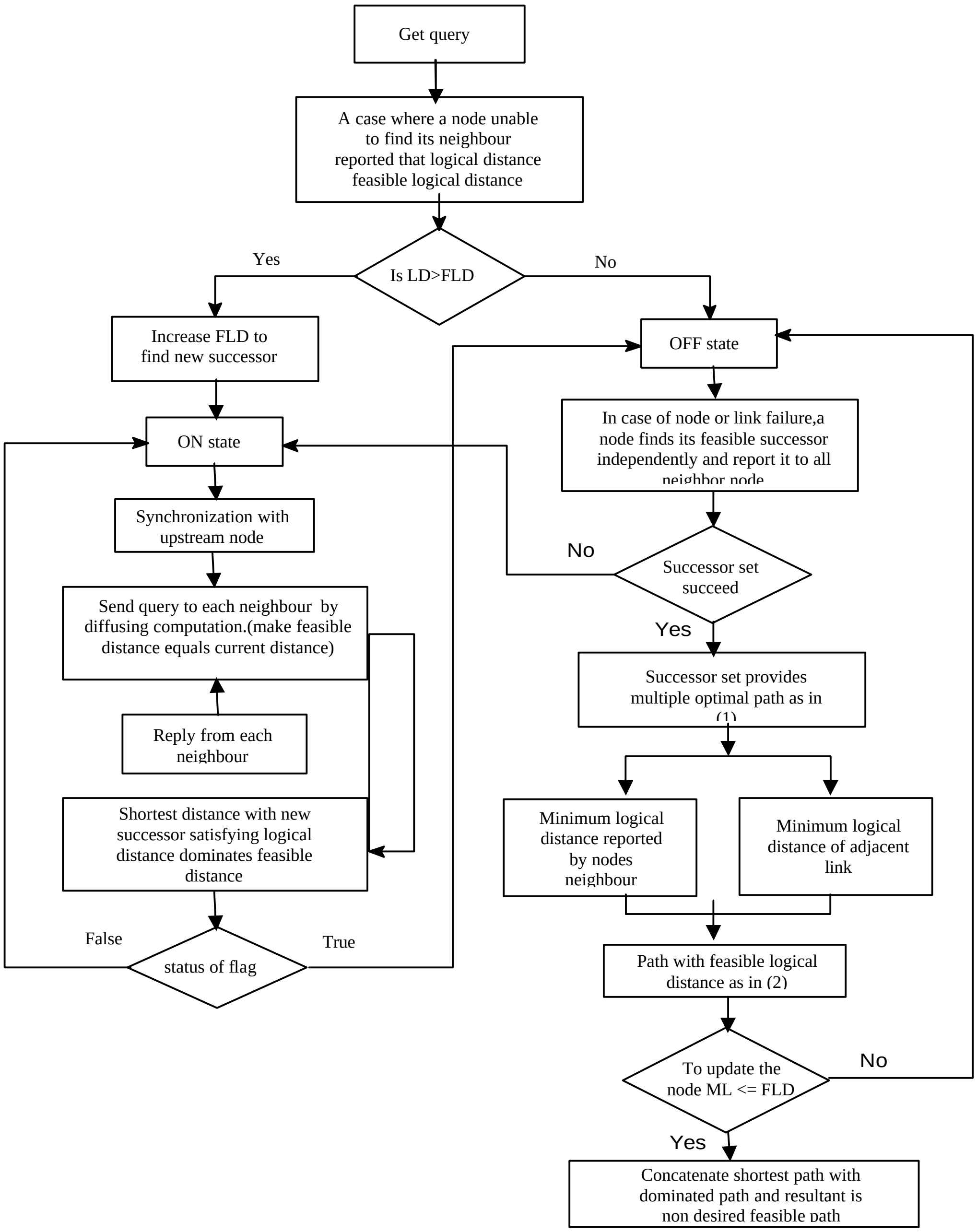


Fig. 1: Flowchart of proposed algorithm

from this, node 'i' refreshes the shortest distance of each feasible path maintained for 'j' and sends neighbors updates if any change occurs and (1) and (2) makes OMCR loop-free QoS routing algorithm. Figure 1 shows flowchart of our algorithm.

**Optimization:** The total number of routing entries for node 'j' maintained at each node forms a directed graph rooted at 'j', which is a sub graph of network G and denoted by $SG_j$ for link $\{l_{i,k} | k \in S_{ij} \text{ for } \forall i \in V\}$. If routing converges correctly, $SG_j$ is a directed acyclic graph in which each node can have multiple successors for node 'j'. At any point of time, multiple $SG_j$ can exit for the destination 'j' to achieve routing optimization, OMCR constructs $SG_j$ in a way that path with shortest distance for destination 'j', is always maintained according to (1) and (2). The paths computed between node 'i' and 'j' are called 'feasible shortest path', denoted by $FSP_{ij}$ and at least one of the path has the minimum shortest distance for 'j'. OMCR sends shortest distance only amongst neighboring nodes, like Distributed Bellman-Ford (DBF) algorithm (Huston *et al*., 2007) and eliminating expensive routing overhead by distributing link-state information throughout the network. Our algorithm may optimize function f(p) and maintain 'x' feasible paths for each destination and hence able to support multi-constrained path selection. The optimization function used is a combination that considers each link-weight component equally which is defined by $f(p)=\sum\frac{W_i}{k}$ where 'k' is number of constraints.

**Network state information:** The underlying network is dynamic and it is essential to update the dynamics of network. It may be the case that node 'i' is unable to find neighbor node 'k' that has reported a shortest distance that is better than the shortest feasible distance maintained at node 'i'. When this happens node 'i' to choose a new set of successors but all nodes whose shortest distance for 'j' involve node 'i' have incorporated that update in their computation of feasible shortest distance. This can be achieved by coordinating node 'I' with all upstream nodes that use node 'i' in their feasible shortest path calculation for destination 'j'. OMCR does spread-out calculation (Aceves, 1993) to achieve this and two state of operation involved namely ON and OFF state. Nodes in OFF state behaves much like a distributed Bellman-Ford algorithm which means that nodes will simply calculate the shortest distance without coordinating with other nodes.

If none of the nodes resulted in an optimal path for the destination 'j' the node may switch to ON state and it increase the feasible shortest distance so that it can coordinate and synchronize to get a new set of successors. By sending 'query' to neighbors that reports the desired shortest distance for destination 'j'. Node '?' returns to OFF state if at least one of the newly elected successor by Eq. 1, provides the feasible shortest distance for destination 'j'.

**Optimization function:** Optimization function f(p) is either policy or application oriented since there is no absolutely better or best routing optimization metric for a given network and therefore specify the properties that an optimization function must have, rather than specifying a specific function, so that diversified routing optimization policies can be defined and implemented and convergence is also arrived quickly if f(p) is properly chosen. As long as shortest distance of a path cannot decrease when the path is extended and the order between two paths must be preserved when they are prefixed or suffixed by a common third path, within the finite time after the last link-state change occurs in the network OMCR converges and maintains the optimal path for each destination. In QoS routing 'sub-optimal property' is not held and hence it is required that the above said two condition should be intact. Time complexity is time to converge after a single change in the network. Communication complexity is the amount of messages required to propagate this change before all nodes that run OMCR integrate it and update their routing Tables accordingly. The computation complexity is the time taken for a node to find process distance vectors regarding a particular destination and it is $O(x|N^i|)$ where $|N^i|$ the number of neighbors of node 'i' and 'x' is the number of routes. The computation complexity is reduced to $O(|N^i|)$ ,if there exists only one single shortest path.

**Example:** A typical topology is shown in Fig. 2 example topology. Feasible path 'p' computed for destination 'k' and weight $W_p$ should also be maintained, because we need to verify $W_p$ whether feasible path can be obtained within the delay bounds. Nodes are labeled with feasible path p(x,y) for the destination 'y' and edges are labeled with their link weights that consist of cost and delay. Distance vectors are propagated from destination and propagate to source through upstream nodes. At node 'e', two paths (c,y) and (f,y) are propagated from neighbor 'c' and 'f' respectively and both have path weight (1,1) $pw_{ij}(m) = pw_{kj}(m) + w_m$ for additive metric where m = 1, 2, ..., number of constraints. By applying above equation at node 'e' we get (e, c, y) = (3, 4) and (e, f, y) = (5, 4). Out of these two we select (e,c,y) since this is outperforming the other. Node 'b' also behaves similarly and obtained two paths (b, c, y) = (2, 3) and (b, f, y)=(3, 6). Since (b, c, y) is better than the other it is selected and propagated to upstream neighbors. Same treatment is applied to nodes 'a' and 'd' and obtain four paths namely (a, e, f, y) = (9, 6), (a, b, c, y) = (8, 8), (a, g, c, y) = (6, 9), (a, e, c, y) = (7, 6) and only (a, g, c, y) and (a, e, c, y) are selected since they provide least feasible paths to the destination. Similar process performed at node 'd' to get the following paths: (d, b, c, y), (d, e, f, y), (d, b, f, y) and (d, h, f, y). Out of these paths (d, h, f, y) = (3, 7) and (d, e, c, y) = (4, 6) are selected and propagated to upstream nodes. Source node 'x' now has four feasible paths to destination 'y' as shown in the Fig. 3 Feasible paths between node 'x' and node 'y'. The four paths are (x, a, g, c, y), (x, a, e, c, y), (x, d, h, f, y) and (x, d, e, c, y).

**Lemma:** OMCR does not create any closed paths/loops while determining the feasible path.

OMCR stays in 'OFF' mode as long as the shortest distance to destination remain unchanged or getting reduced. But when the shortest distance increases to a particular destinations, for which nodes send out queries and transit into 'ON' state i.e., a node that runs OMCR synchronizes with upstream nodes and raises its feasible shortest distance up to a sufficient value such that another set of successors can be obtained and return to 'OFF' mode.

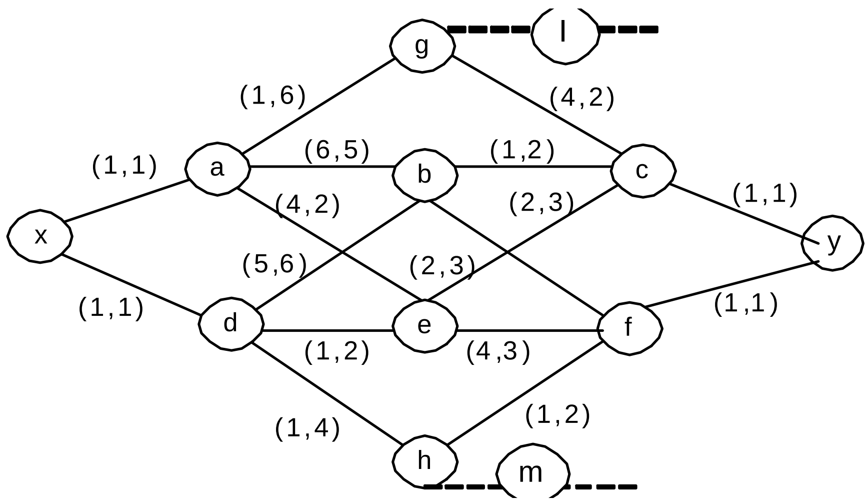


Fig. 2: Example topology

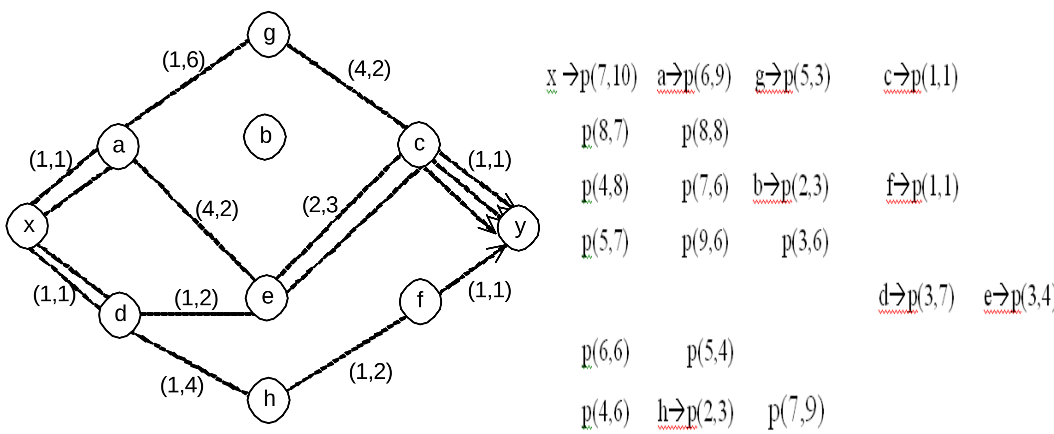


Fig. 3: Feasible paths between node 'x' and 'y'

It can be seen that Feasible Shortest Distance (FSD) used by node 'i' to 'j' is lesser than shortest path distance for 'j' that node 'i' reports to all its neighbors (RSD) for ∀t. If node 'i' switches to 'ON' mode by sending queries out and a neighbor receives a query at time $t_1 > t_n$ and if nodes are updated regularly then $FSD_{ij}^{(t)} \le SD_{ji}^{k(t)}, t \in (t_n, t_1)$. Otherwise a new active phase follows immediately at time $t_2 > t_1$ queries are sent out and at $t_3 > t_2$ all replies are received at node 'i'. At any point of time, query and reply are not getting overlapped and hence OMCR does not form any loop at any given point of time.

## RESULTS AND DISCUSSION

**Simulation results:** In this section, we present some of our experimental results which are carried out in a discrete-event C++ simulator. We adopted a random graph and Waxman graph (Waxman, 1988) used by many researches (Chen and Nahrstedt, 1998; Yuan, 2002). In random graph, the existence of the link between any two nodes is determined by a constant probability 'pr' where 0<pr<1 while in Waxman topology 'pr' is defined as $pr = \alpha e^{-d/\beta L}$, where '$\alpha$>0, $\beta$<1 where 'd' is the distance between two nodes and L is the maximal inter nodal distance in the graph and geographical distance is also taken into account. For all configurations, each link weight component is uniformly distributed (Yuan, 2002). Each source-destination pair is randomly chosen from the networks. We consider only additive routing metrics so that we can compare our performance with other multi-constrained path algorithms.

We compare OMCR with Distributed Bellman-ford (DBF) (Huston *et al*., 2007) algorithm which is based on Distance Vector (DV) routing Link State (LS) routing. We studied establishing routes for all destinations at each node for the first time since nodes are 'up'. We further

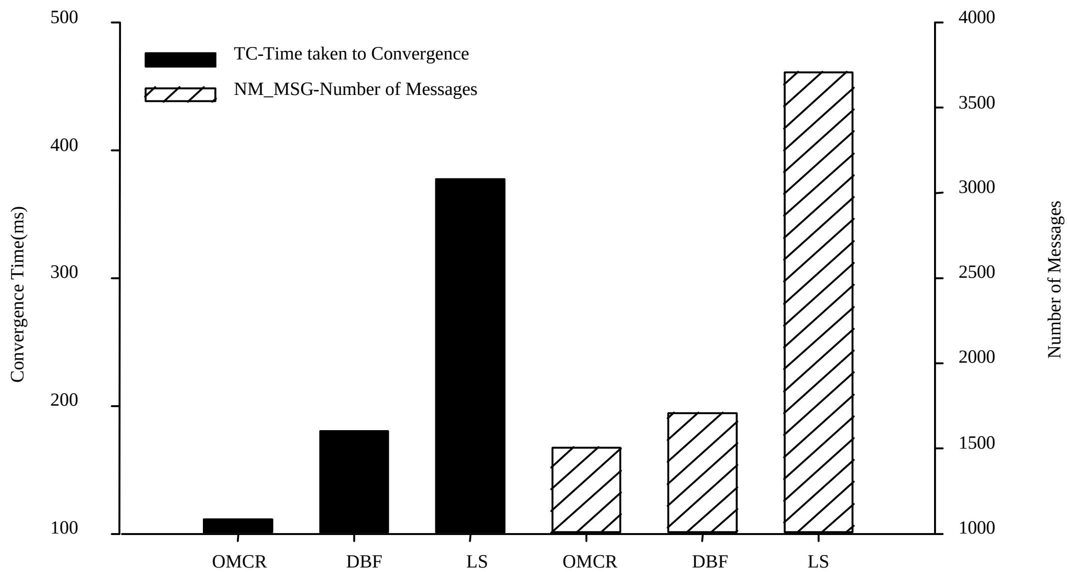


Fig. 4: Establishment of routes

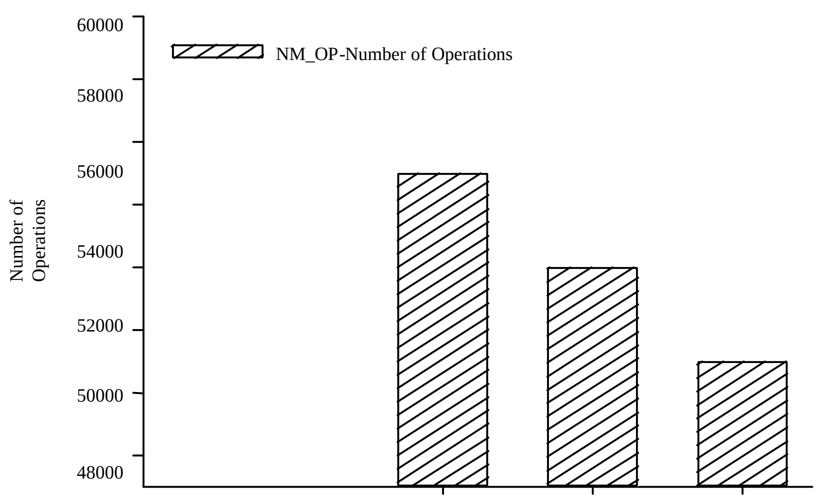


Fig. 5: Number of operations to establish routes

analyzed single link failure, node failure and finally a sequence of link failures. We simulated link or node failures since there are worst case scenario that may cause shortest distance to increase or network partition. Convergence time of OMCR is comparatively less as shown in Fig.4.

We analyzed 'number of operation (NM_OP) which denotes iterations of the main loop in its implementation i.e., sum over all nodes in the network. Then the number of messages (NM_MSG) which is the total number of messages sent over all links. Figure 5 shows the number of operation required to establish a route. Even though the number of operations are little higher it forms a loop-free route from source to destination. These two metrics are measured since the start of the simulation until all nodes stop updating their tables as shown in Fig. 6 Single link failure. Finally, we measured the time to converge the network which is the time since the start of an event until all nodes stop updating their routing tables.

In case of sequence of link failures, multiple links are randomly selected to go down at different point of time. The total number of operations required in case of link failure is shown in Fig. 7.

Better convergence time is achieved in OMCR compared with its counterparts as shown in Fig. 8 Single node failure Vs convergence time. The overhead on

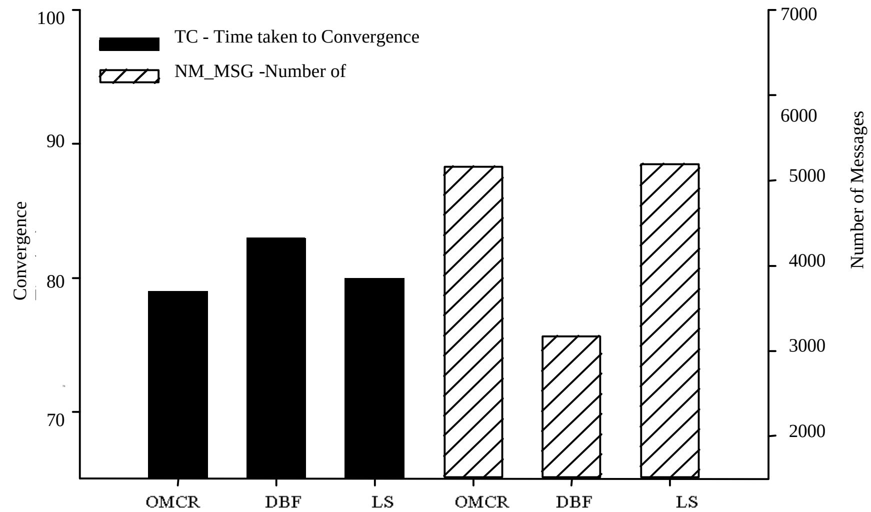


Fig.6: Single link failure

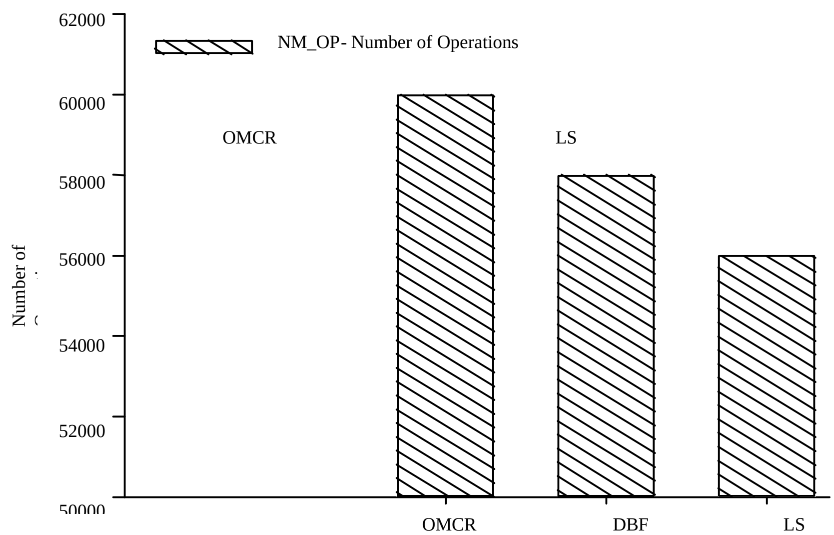


Fig.7: Single link failure

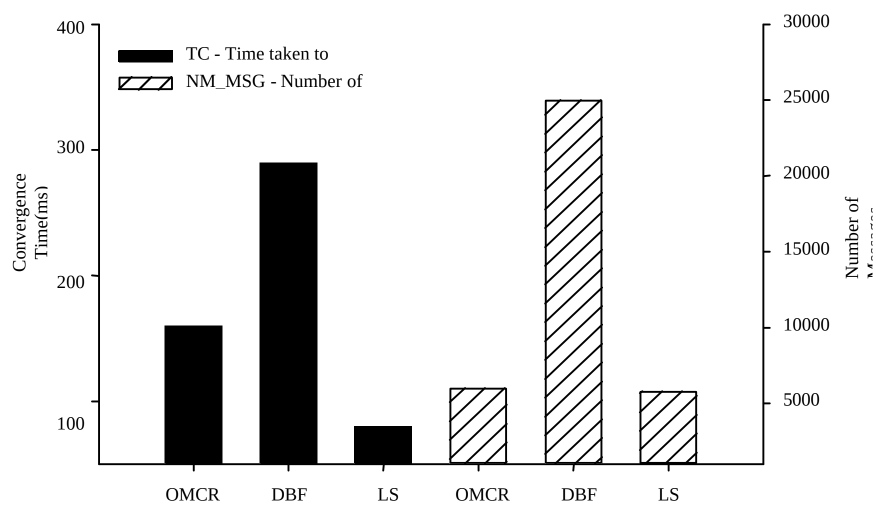


Fig.8: Single node failure Vs convergence time

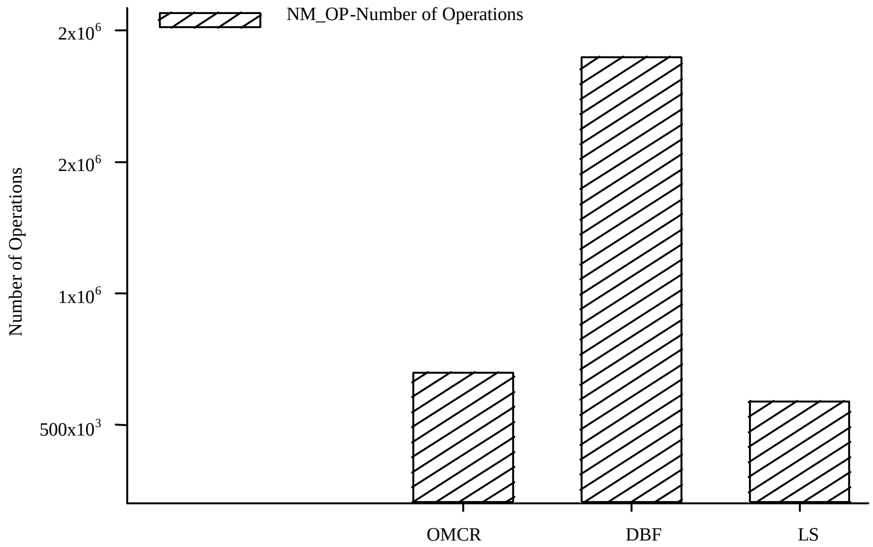


Fig.9: Single node failure Vs number of operations

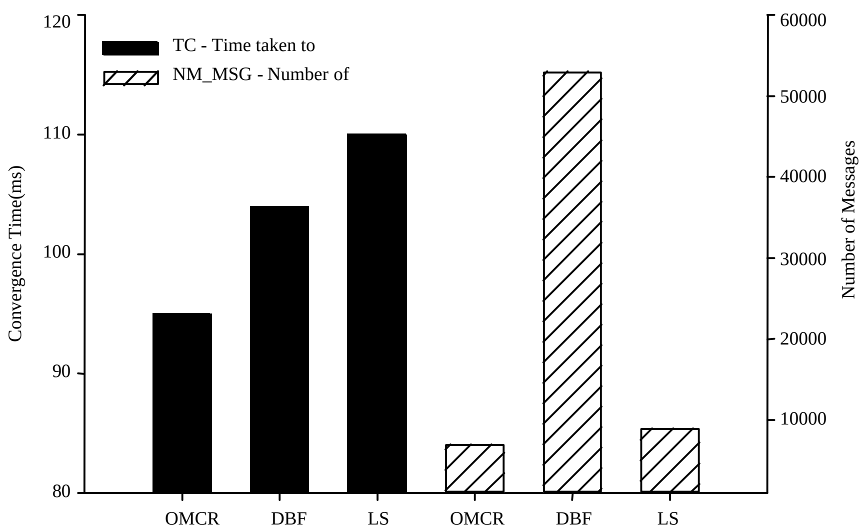


Fig.10: Multiple link failure Vs convergence time

update messages will be reduced significantly if multiple vectors are allowed to send. Compared to link state routing, the number of update messages are less in OMCR as depicted in Fig. 9; this is obvious that in LS routing the whole network need to be updated whenever topology changes. OMCR behaves much like distance vector routing while a link fails and no network disastrous events are reported, since the degree of nodes are maintained more than one, possibilities of network isolation is drastically reduced.

Due to count-to identify problem, the number of messages exchanged is quite high in case of node failure and sequence of link failures in distance vector routing compared to either OMCR or link state routing as in Fig. 10 Multiple link failure Vs Convergence time. If all adjacent links are failed of a particular node at any point in time, again distance vector routing behaves poorly owing to the same reason.

In case of link state routing, metrics such as number of operations and convergence time are above than that of OMCR and it may be understood that while a link or node fails, more number of updates need to be sent out to keep nodes updated about the current network state and the routing tables also need to be refreshed more frequently in order to cope with latest changes. The more link failures occur, the worse link state routing performs as

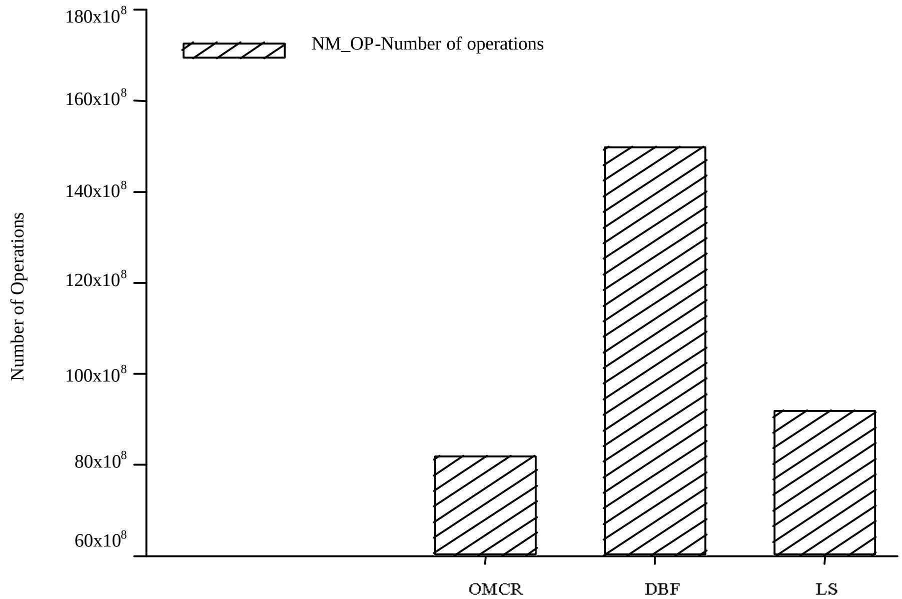


Fig.11: Multiple link failure Vs number of operations

Table 1: Comparison of success rates

| Routing requests routed | | | | | | |
|---|---|---|---|---|---|---|
| SR (%) | JSP | SR (%) | FPATC | SR (%) | OMCR | Total routing requests |
| 81.00 | 81 | 91.00 | 91 | 95.00 | 95 | 100 |
| 86.00 | 172 | 94.00 | 188 | 97.00 | 194 | 200 |
| 89.33 | 268 | 93.66 | 281 | 97.33 | 292 | 300 |
| 89.50 | 358 | 94.00 | 376 | 97.75 | 391 | 400 |
| 89.80 | 449 | 93.60 | 468 | 97.00 | 485 | 500 |
| 93.50 | 561 | 94.16 | 565 | 96.66 | 580 | 600 |
| 94.14 | 659 | 95.71 | 670 | 97.57 | 683 | 700 |
| 93.50 | 748 | 94.75 | 758 | 96.87 | 775 | 800 |
| 93.33 | 840 | 94.55 | 851 | 96.88 | 872 | 900 |
| 93.10 | 931 | 94.90 | 949 | 96.90 | 969 | 1000 |

in Fig. 11 Multiple link failure Vs Number of operations. This shows that OMCR can provide differentiated QoS provisioning with better or comparable routing complexity than shortest-path routing protocols designed for best-effort traffic.

We compare Rate of Success (RS) of OMCR in Table 1 Comparison of success rates with other algorithms that require complete-state information namely Feasible Path Algorithm with Two Constraints (FPATC) Korkmaz *et al*. (2000) and variation of Jaffe algorithm (JSP), where a shortest path algorithm with reference to the aggregated link-cost function. For our calculation, we define the rate of success as the ratio of routing requests routed to total routing requests received at any given point of time We found that performance of Jaffe's algorithm is not coping with other algorithms in all circumstances which derive that single aggregated metric may not be the better approach for constrained path computation. The performance of FPATC is good in term of success rate but handles only two additive constraints and running time cannot be ascertained if the number of constraints are more, whereas OMCR solves general K-constrained MCP problems that includes both additive and minimal constraints.

## CONCLUSION

Optimal path selection subject to multiple constraints is an NP complete problem which can be addressed through heuristics and approximation algorithms. OMCR is a QoS routing algorithm uses distance vectors to solve multi constrained path problems with loop-free paths and does not require storing the global network state. We also addressed optimization issues in routing. Our experiments show that OMCR outperforms the shortest path routing algorithm that are being used in current Internet environment with regard to network overhead and routing complexity. Further, our experiments reveal that having global network state at each node is not always a good approach in practice; it is particularly true when network state changes quite frequently which usually occurs in QoS routing. Convergence time of our algorithm is comparatively less. Finally our algorithm achieves a better routing success ratio while comparing to other routing schemes in link state segment.

## REFERENCES


Aceves, J.J.G.L., 1993. Loop-Free routing using diffusing computations. IEEE/ACM. Trans. Networking, 1: 130-141.

Chen, S. and K. Nahrstedt, 1998a. Distributed QoS routing with imprecise state information. Proceedings of the 7th International Conference on Computer Communications and Networks, October 15-15, 1998, IEEE, New York, USA., ISBN: 0-8186-9014-3, pp: 614-621.

Chen, S. and K. Nahrstedt, 1998b. On finding multi-constrained paths. Proceedings of the 1998 IEEE International Conference on Communications, June 7-11, 1998, IEEE, New York, USA., ISBN: 0-7803-4788-9, pp: 874-879.

Ghosh, D., V. Sarangan and R. Acharya, 2001. Quality-of-service routing in IP networks. IEEE. Trans. Multimedia, 3: 200-208.

Hutson, K.R., T.L. Schlosser and D.R. Shier, 2007. On the distributed Bellman-Ford algorithm and the looping problem. Informs J. Comput., 19: 542-551.

Jaffe, J.M., 1984. Algorithms for finding paths with multiple constraints. Netw., 14: 95-116.

Korkmaz, T., M. Krunz and S. Tragoudas, 2000. An efficient algorithm for finding a path subject to two additive constraints. Proceeding of the ACM SIGMETRICS Conference on Performance Evaluation Review, June 18-21, 2000, ACM, New York, USA., ISBN:1-58113-194-1, pp: 318-327.

Mieghem, P.V. and F.A. Kuipers, 2004. Concepts of exact QoS routing algorithms. IEEE/ACM. Trans. Networking, 12: 851-864.

Mieghem, P.V., H.D. Neve and F. Kuipers, 2001. Hop-by-hop quality of service routing. Comput. Netw., 37: 407-423.

Neve, H.D. and P.V. Mieghem, 2000. TAMCRA: A tunable accuracy multiple constraints routing algorithm. Comput. Commun., 23: 667-679.

Prakash, P.S. and S. Selvan, 2008. A feasible path selection QoS routing algorithm with two constraints in packet switched networks. Intl. J. Comput. Inf. Sci. Eng., 2: 129-135.

Smith, B.R. and J.J.G.L. Aceves, 2004. Efficient policy-bsed routing without virtual circuits. Proceedings of the 1st International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, October 18-20, 2004, IEEE, New York, USA., ISBN: 0-7695-2233-5, pp: 242-251.

Sobrinho, J.L., 2001. Algebra and algorithms for QoS path computation and hop-by-hop routing in the internet. Proceedings of the IEEE 20th Annual Joint Conference on Computer and Communications Societies, April 22-26, 2002, IEEE, New York, USA., ISBN: 0-7803-7016-3, pp: 727-735.

Wang, J. and K. Nahrstedt, 2002. Hop-by-hop routing algorithms for premium-class traffic in diffserv networks. Proceedings of the IEEE 21st Annual Joint Conference on Computer and Communications Societies, June 23-27, 2002, IEEE, New York, USA., ISBN: 0-7803-7476-2, pp: 705-714.

Wang, Z. and J. Crowcroft, 1996. Quality-of-service routing for supporting multimedia applications. IEEE J. Selected Areas Commun., 14: 1228-1234.

Waxman, B.M., 1988. Routing of multipoint connections. IEEE J. Selected Areas Commun., 6: 1617-1622.

Yuan, X., 2002. Heuristic algorithms for multiconstrained quality-of-service routing. IEEE/ACM. Transac. Networking, 10: 244-256.