New Localized Call Admission Control Algorithms in Communication Networks with Quality of Service Constraints

Abdulbaset H. Mohammad

School of Computing, Informatics and Media University of Bradford, United Kingdom

Summary

Localized Quality of Service (QoS) routing has been recently proposed for supporting the requirements of multimedia applications and satisfying QoS constraints. Localized algorithms avoid the problems associated with the maintenance of global network state by using statistics of flow blocking probabilities. Using local information for routing avoids the overheads of global information with other nodes. However, the localized QoS routing algorithms were only path selection routing algorithms and this leads to them accepting every incoming flow that can be physically accommodated. This paper introduces call admission control algorithms with localized QoS routing in order to maximize the connections that network accepts and improve the network resource utilization. Simulations of various network topologies are used to illustrate the performance of the algorithms. We compare the performance of the algorithms against the Credit Based Routing (CBR) algorithm under various ranges of traffic loads.

Keywords:
Localized QoS routing algorithms, Call admission control algorithms, Performance Evaluation

1. Introduction

The concept of QoS capabilities is a challenging task. A major concern is that in the data packets belonging to the same flow may route traffic among different paths to the destination. These difficulties can be overcome in different network topologies dynamically setup paths with QoS guarantees. QoS routing is recognized an efficient method to find a path of providing guaranteed QoS to the Internet. There are a wide variety of proposed solutions to the problem of selecting a path with specific QoS requirements, a comprehensive survey can be found in [1]. Such routing algorithms depend on global network state information and need to be exchanged periodically using link state information in order to make routing decisions.

High levels of exchange may incur large communication and processing overheads [2]. The localized QoS routing is proposed [3] [4] attempts to overcome the problems associated with the maintenance of the global network state information by making routing decisions based solely on the information collected locally at each source. In localized QoS routing schemes each source node has a predetermined set of candidate paths to each of the destinations.

This paper introduces call admission control algorithms with localized QoS routing in order to maximize the connections that network accepts, distribute the load throughout the network and thus improve the network resource utilization. Simulations of various network topologies are used to illustrate the performance of the algorithms. We compare the performance of the algorithms against the Credit Based Routing (CBR) algorithm under various ranges of traffic loads.

2. Related Algorithms

Localized Quality of Service routing has recently been introduced as a new approach in the context of QoS routing. To the best of our knowledge our simulation study is the first that considers call admission control algorithms with localized QoS routing on the performance of the QoS traffic guaranteed. The main localized quality of service routing algorithms are:

- *Localized Proportional Sticky Routing Algorithm*

The localized proportional sticky routing algorithm (PSR) [3] was the first localized QoS routing scheme used in the context of computer networks. The basic idea behind the PSR approach assumes that route level statistics, such as the number of flows blocked, is the only available QoS state information at a source and based on this information the algorithm attempts to proportionally distribute the traffic load from a source to a destination among the set of candidate paths, according to their flow blocking probability. With this scheme each source node needs to maintain a set of candidate paths R. A path is based on flow blocking probability and the load is proportionally distributed to the destination among the predefined paths. In PSR there are minimum hop paths R^{min} and alternative paths R^{alt}, where $R = R^{min} \bigcup R^{alt}$.

The PSR algorithm can be viewed as operating in two stages: proportional flow routing and computation of flow proportions. The scheme proceeds in cycles of variable lengths which form an observation period. During each

---

Manuscript received November 5, 2010
Manuscript revised November 20, 2010

cycle along a path r, any incoming connection request can be routed among paths selected from a set of eligible paths R^{alt}, which initially may include all candidate paths. A candidate path is ineligible depending on the maximum permissible flow blocking parameter γ_r, which determines how many times this candidate path can block a connection request before it becomes ineligible.

For each minimum hop path, γ_r is set to $\hat{y}$, which is a configurable parameter, whereas the alternative path γ_r is dynamically adjusted between 1 and $\hat{y}$. When all candidate paths become ineligible a cycle ends and all parameters are reset to start the next cycle. An eligible path is finally selected depending on its flow proportions. The larger the flow proportions, the larger chances for selection.

At the end of the observation period, a new flow proportion α_r is computed for each path in the candidate path set, based on its observed blocking probability b_r. After each observation period the minimum hop path flow proportions are adjusted to equalize their blocking probability $(\alpha_r.b_r)$. For the alternative paths, the minimum blocking probability among the minimum hop paths b* is used to control their flow proportion. That is, for each $r \in R^{alt}$, if $b_r<\psi b^*$, γ_r=min (γ_r+1, $\hat{y}$). If $b_r> b^*$, γ_r=max (γ_r-1, 1), where ψ is a configurable parameter to limit the 'knock-on' effect [3] under system overloads. Note that $\gamma_r \geq 1$ ensures that some flows are routed along alternative paths to measure their quality.

- *Localized Credit Based Routing Algorithm*

The Credit Based Routing (CBR) [4] algorithm uses a simple routing procedure to route traffic across the network. The CBR scheme performs routing using crediting scheme for each candidate path that rewards a path upon flow acceptance and penalizes it upon flow rejection. The larger path credits, the larger chances for selection. The CBR algorithm keeps updating each path's credit upon flow acceptance and rejection and it does not compute a flow proportion. It is also keeps monitoring the flow blocking probabilities for each path and conveys the data to the credit scheme to use it in path to path selection. A set of candidate paths R between each source and destination is required in the CBR algorithm. Like PSR, CBR predetermined a minimum hop set $R^{\min}$ and an alternative paths set R^{alt} where $R = R^{\min} \cup R^{alt}$. CBR selects the largest credit path P.credits in each set, minimum hop paths set $R^{\min}$ and alternative paths set R^{alt} upon flow arrival. The flow is routed along the minimum hop path that has the largest credit $P^{\min}$ which is larger than the alternative path that has the largest credits P^{alt}; the flow is routed along an alternative path using this formula (1):

$$P^{\min}.credits \geq \Phi \times P^{alt}.credits \text{ , where } \Phi \leq 1 \qquad (1)$$

Φ is a system parameter that controls the usage of alternative paths. The CBR uses blocking probability in crediting schemes to improve the performance of the algorithm. The path credits are incremented or decremented upon flow acceptance or rejection using statistics of the path blocking probability.

However, CBR uses a MAX_CREDITS system parameter to determine the maximum attainable credits for each path by computing the blocking probability.

$$0 \leq Credits \leq MAX\_CREDITS \qquad (2)$$

CBR algorithm records rejection and acceptance for each path and uses a moving window for a predetermined period of M connection requests. It uses 1 for flow acceptance and 0 for flow rejection, dividing the number of 0's by M to calculate each path blocking probability for the period of M connection requests.

Although the first localized QoS routing algorithm was Proportional Sticky Routing (PSR) [3], it has subsequently been shown that CBR outperform PSR [4] [5]. The CBR will thus subsequently be used for benchmarking.

3. Call Admission Control based on Bandwidth as QoS Metric

This paper distinguishes between flow admission control and higher-level admission control. Flow admission control can be defined in the following way: a flow is routed over a path as long as it passes the admission control and resource reservation of each intermediate node along the path. While this type of admission control is required to control flow admission at each node [6], efficiency of QoS routing may require an additional layer of admission control. Higher-level admission control would consider the resource requirement of each flow in relation to the available resources along a path, in order to determine whether it is profitable overall to admit the flow. Thus a flow may be rejected even if there is a feasible path to route the flow [6] [7].

- *The Proposed Algorithm*

A new call admission control was used with localized congestion avoidance source routing where the source nodes take routing decision to route traffic from the source to the destination.

The incoming flow can be routed among explicit paths selected from a set of candidate paths. It is assumed that signalling and resource reservation are used to make a path for each connection request. The signalling process starts at the source node by sending a setup message along the selected path. Each intermediate node performs an admission test to see whether the outgoing link has sufficient residual bandwidth for the new flow. If the link can accommodate the new flow, the requested bandwidth is reserved for that flow then the message is forwarded to the next link. The flow is admitted if all links can support the flow, otherwise the flow is rejected and a failure message is propagated back to the source node.

However, the localized PSR and CBR and other localized QoS routing algorithms were only path selection routing algorithms and this leads to them accepting every incoming flow that can be physically accommodated. Our algorithm differs from previous localized algorithms since call admission control algorithms proposed to manage the incoming flows.

The pseudo code for the CAC algorithm as given in Figure 1, as follows:

```
Initialize ( )
Set P.Avg=1,     P   R
CAC( )
1. if P.Avg=0     P   R
2. Set P.Avg=1,     P   R
3. Set P=max{P.Avg: P   Rmin}.
4.   If All Candidate Paths are Congested
     // Requested Bandwidth > Residual Bandwidth
        Block the flow
5.   Else
         Route the flow along path P
4 If flow routed through congestion path
5.            Change path P
6. else
7. Route flow along path P
      Predefined Period:
8.   if flow rejected
        Set Congestion Path
```

Figure 1 The pseudo code for the CAC algorithm

Localized routing requires a predetermined set of candidate paths R. The main characteristic is that it is associated with every path P in the candidate path set. The algorithm periodically advertises the congestion state by updating the blocked link in each blocking path request.

Upon flow arriving at the source node, the signalling process starts to select a path to route the flow. If the link cannot accommodate the new flow, the flow is rejected and the algorithm stores this congestion state and a failure message is propagated back to the source node. Note that the congestion state updated in each predefined period. The algorithm starts to select paths between the source and destination. If any of the future flow arrivals routed through the congestion path the algorithm replaces the path with a second higher quality path in order to avoid congestion. The algorithm admits the connection if all links along the path can satisfy the congestion state and satisfy the requested bandwidth for the flow.

Upon flow arrival, the source node performs call admission control by blocking the requests that do not satisfy the requested bandwidth. It then prevents the arriving requests from entering the network when all the paths between each source and destination are congested. It does not require a signalling process to route the flow along the network and then propagate the failure message back to the source. It is not a good idea to route traffic along the link of congestion.

4. Call Admission Control based on Delay as QoS Metric

In this Section we assume that the QoS constraint is end-to-end delay as QoS metric. We particularly focus on mean delay and thus the end-to-end delay is the accumulation of delays at each router along the path with intermediate nodes. Although this metric can not guarantee instantaneous delay required in some real time applications it provides a useful metric in which their applications are required. Using instantaneous delay as a QoS metric is therefore meaningless and required to be represented by a random variable which the distribution changes instantaneously at the packet level.

- Delay Call Admission Control

Localized QoS constraint presented for traffic requiring delay guarantees However, these QoS networks do not guarantee QoS traffic because of uncontrolled admission is not acceptable for QoS networks, as a newly admitted connections may jeopardize the QoS of an existing connections. Admission of new connections therefore must be controlled carefully in the QoS networks to protect the connections currently in progress. The concept of controlling the admission of new connection is known as delay call admission control. Note that this situation is different from using bandwidth as the QoS metric since mean delay is an additive QoS metric. In order to overcome this problem we introduce a delay call admission control with the proposed localized routing algorithms to efficiently manage the resources among existing and new flows, we examine the overall performance while aims to provide the end-to-end QoS guarantees.

- The Proposed Algorithm

A new localized Average Acceptance routing algorithm with delay admission control DABR was proposed to make routing decisions. We also modify the CBR [4] so that uses delay instead of bandwidth to perform routing decisions. The intelligent of our algorithm is compared with modified CBR (DCBR) algorithm under different network topologies. The DABR scheme with delay constraint is imposed not only for the new incoming flows but also for already admitted flows.

The connection signalling in DABR starts when incoming flow arrives at the source, the source node sends a setup message along the selected candidate path. The message stores the delay over the ongoing link and each intermediate node of the routing path P_{s-d} performs an admission test for the outgoing link to accumulate the outgoing link delay to the previous delay.

If the delay that the message experiences is less than the QoS delay and this value of delay is less than the accumulated delay of the existing flows, the delay is reserved for that flow and the message is forwarded to the next node. The flow is accepted if the actual delay experienced in the path routing is less than the QoS delay and this path does not jeopardize the QoS of any existing path, which means that the end-to-end delay of the routing path satisfies the requested QoS.

The end-to-end delay of the routing path is the accumulation of all delays in P_{s-d} is given as:

$$D\left(P_{s-d}\right)=\sum_{l_{m-n}\in P_{s-d}} d\left(l_{m-n}\right) \tag{3}$$

Where l_{m-n} is the links along the source destination path.

Given a delay constraint QoS_Delay, the problem of delay constraint for localized QoS routing a path will accepts only where the sum of delays does not exceed QoS_Delay.

$$D\left(P_{s-d}\right)=\sum_{l_{m-n}\in P_{s-d}} d\left(l_{m-n}\right)\le QoS\_Delay \tag{4}$$

If we consider the call admission control to guarantee the requested QoS of the existing flows:

$$D\left(NewP_{s-d}\right)=\sum_{l_{m-n}\in P_{s-d}} d\left(l_{m-n}\right)\le QoS\_Delay \tag{5}$$

$$D\left(ExistingP_{s-d}\right)=\sum_{l_{m-n}\in P_{s-d}} d\left(l_{m-n}\right)\le QoS\_Delay \tag{6}$$

The information statistics regarding flow acceptance or rejection is conveyed to source node to make routing decision. The pseudo code for DABR algorithm is described in Figure 2.

DABR requires every node to maintain a predetermined *set of candidate paths R* for each possible destination.

When the flow is accepted along the selected path, its acceptance rate is accordingly updated and P.Rate is incremented by the value that corresponds to it. On the other hand, if the flow is rejected (line 13) its acceptance rate is accordingly updated and P.Rate is decremented by the value that corresponds to its acceptance rate, as shown in the pseudo code.

Since the DABR algorithm continuously monitors the acceptance rate, it records flow data (acceptance and rejection) for every path and uses a sliding window with a predetermined period to calculate the path's acceptance rate.

```
Initialize
      Set  P.Rate=1, $\forall$ $P \in R$
DABR
    1.  if P.Rate=0 $\forall$ $P \in R$
    2.    set P.Rate=1, $\forall$ $P \in R$
    3.  $P^{min}$=max{P.Rate: $P \in R^{min}$}.
    4.  $P^{alt}$ =max{P.Rate  $P \in R^{alt}$ }.
    5.  if($P^{min}$.Rate>= $\Phi \times P^{alt}$.Rate).
    6.      set  P=$P^{min}$
    7.  else
    8.  set   P= $P^{alt}$
    9.  route flow along path P.
    10. if Sum{L.Delay:$L \in P$} $\le$ QoS_Delay
  &&Sum{L'.Delay+L.Delay} $\le$ QoS_Delay
  For (Exsiting P, , $\forall$ $P \in R$
    11. if flow accepted
    12. UpdatePath'sAcceptanceRate(1)
    13. Compute P.Rate.
    14. else
    15. UpdatePath'sAcceptanceRate(0)
    16. Compute P.Rate.
```

Figure 2 the DABR algorithm

For a period of M, the acceptance rate of every path will be calculated using the most recent M flow data. This is can be implemented easily using a sliding window with fixed size M. The new value is added to the beginning of the list after removing the oldest value from the list.

Figure 2 is the DABR pseudo code for updating the acceptance rate of a path. As an example, let S= $\{\underline{1},1,1,1,0\}$ represent the information regarding acceptance and rejection of the last M =5 flows, where 0 indicates flow rejection and 1 indicates flow acceptance, then the acceptance rate will be 4/5. Now, if another flow is rejected, then the oldest element (leftmost position) will be deleted from S, and replaced by the data from the last flow, i.e. $S=\{\underline{1},1,1,0,0\}$ then the acceptance rate will be 3/5, and updated accordingly. In contrast to the previous localized schemes, which indirectly reflect the quality of the path by the addition or subtraction of the credits criterion, the DABR computes the acceptance rate directly.

5. Performance Evaluation

This section evaluates the performance of the proposed localized call admission control schemes based on bandwidth and delay QoS metrics. We first describe a simulation model and then compare the performance using a flow blocking probability as a performance metric.

5.1 Simulation Model

We implemented the scheme based on an event driven simulator OMNET++ [8] and conducted extensive simulations to test the performance. Using the predetermined algorithms the simulation performs path selection, resource reservation, and admission control at flow level.

We used a familiar ISP topology which is widely used in simulation studies [9]. In addition we investigate random and regular topologies. The random topologies were generated on top of the Brite generator [10] using Waxman's model [11]. Table 1 lists the most important characteristics of the topologies used in our simulation experiments.

Topology	Nodes	Links	Avg. Path Length
ISP	32	108	3.177
RANDOM 32	32	122	2.416
SCALE-FREE	32	122	2.4274

5.2 Traffic Model

All the links are assumed to be symmetric, bidirectional and have the same capacity C (C=150 Mbps) in each direction. The topology remains fixed through each simulation experiment; hence, we do not model the effects of link failures. Flows arrive to each source node according to a Poisson distribution with rate $\lambda=1$, and destination nodes are selected randomly by uniform distribution. Flow duration is exponentially distributed with mean value $1/\mu$, while flow bandwidth (QoS requested) is uniformly distributed within a [0.1-2MB] interval. Following [2] [12], offered network load is $\rho = \lambda N \bar{b} \bar{h} / \mu L C$, where N is the number of nodes, $\bar{b}$ is the average bandwidth required by a flow, L is the number of links in the network, and $\bar{h}$ is the mean number of hops per flow, averaged across all source-destination pairs.

he parameters used in the simulation of the CBR algorithm MAX_CREDITS=5, and $\Phi=1$ unless otherwise stated.

Blocking probabilities are calculated based on the most recent 20 flows. Following [13], a set of candidate paths is chosen such that, for each source-destination pair, the candidate path set consists of paths have at most one hop more than the minimum number of hops. In simulation experiments reported here, the 95% confidence intervals were computed and found to be extremely close, such that they were not visible on the graphs with the scales used.

Each result simulates the arrival of 2,000,000 flows and the simulation results are collected after the first 500,000 flows, used as an initialization period.

5.3 Performance Metric

The metrics used to measure the performance of the algorithms are the flow blocking probability and the bandwidth blocking probability

- *Flow blocking probability* is defined as:

$$\textit{Flow}\ \text{blocking probability}=\frac{\text{No of rejected flows}}{\text{No of arrived flows}} \quad (7)$$

5.4 Simulation Results

Figures 1, 2 and 3 observe the performance of CAC algorithm based on bandwidth metric compared with existing CBR algorithm in different topologies.

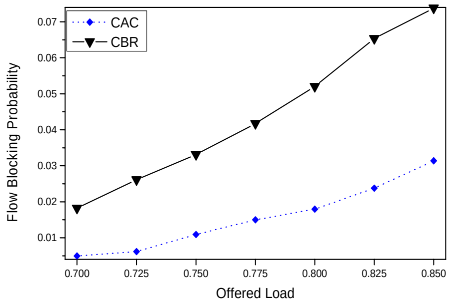

Figure 3 CAC based on Bandwidth metric in RANDOM32 Topology.

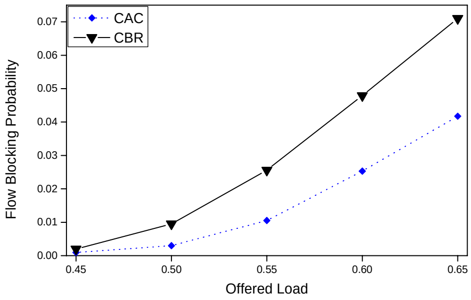

Figure 4 CAC based on Bandwidth metric in ISP Topology.

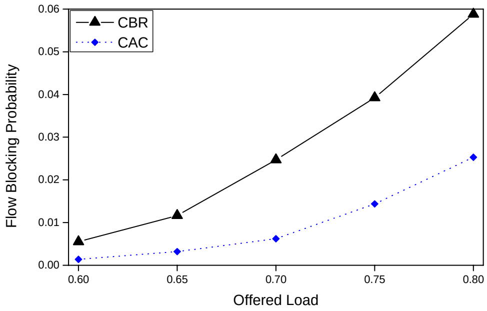

Figure 5 CAC based on Bandwidth metric in Scale-Free Topology.

Figures 3, 4 and 5 compare the performance of CAC algorithm in terms of flow blocking probability plotted against various ranges of loads under different network topologies. CAC algorithm gives better performance comparing with CBR algorithm due to controls the flows entering the network and make routing decision based on the congestion state. The performance under RANDOM32 observed superior performance because it is a dense topology and there are enough candidate paths between each pair of nodes.

The CAC algorithm distributes the load throughout the network since the algorithm uses more likely loaded links to avoid congestion.

Figures 6 and 7 observe the performance of CAC algorithm based on delay metric compared with existing DCBR algorithm in different topologies.

Figures 6 and 7 compare the performance of DABR algorithm in terms of flow blocking probability plotted against various ranges of QoS delay constraints for different network topologies. It can be noted that the algorithm satisfy most flows under large delay constraint (delay constraint=3) as expected since the probability of finding a path that satisfies a large constraint is high and most flows will be accepted.

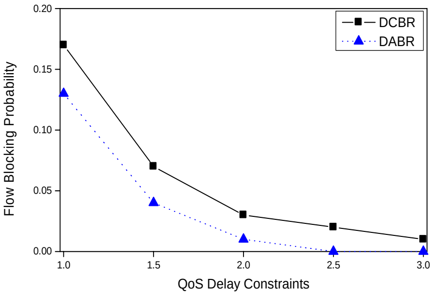

Figure 6 CAC based on Delay metric in RANDOM32 Topology.

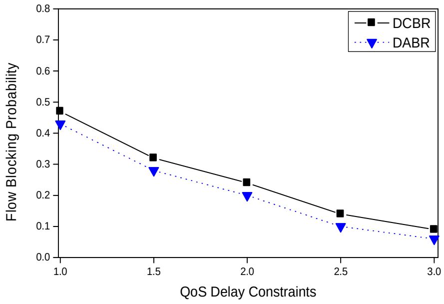

Figure 7 CAC based on Delay metric in ISP Topology.

The performance under dense networks RANDOM32 topology is better than for ISP topology because the algorithm is hard to use scarce resources in the sparsely connected ISP topology as shown in Figure 7.

The call admission control algorithms tend to reduce the overall blocking probability since flows may not then be routed over long paths. This reduces the overhead by minimizing the signalling effort.

4. Conclusion

Unlike the previous localized schemes, which they are only path selection routing algorithms, we introduce the call admission control algorithms based on quality of service constraints integrated with localized QoS routing. We compared the proposed algorithms with localized CBR algorithm. The simulation results show that our call admission control algorithms increase the connections that the network accepts in terms of flow blocking probability. We have demonstrated that the algorithms perform well even the blocked flows at the source by call admission control algorithms are considered with the overall flow blocking probabilities. The algorithms distribute the load efficiently and reduce the overhead by minimizing the signalling overhead.

Acknowledgment

The author would like to express his cordial thanks to the three anonymous reviewers for their valuable comments and useful feedback in evaluating the paper.

References

[1] S. Chen, an K. Nahrstedt, "An Overview of Quality of Service Routing for the Next Generation High Speed Networks: Problems and Solutions", IEEE Network Magazine, 12(1998), pp.64–79.

[2] AS. Shaikh, J. Rexford and K. Shin, Evaluating the Impact Stale Link State on Quality-of-Service Routing IEEE/ACM Transactions on Networking, 9 (2001), pp. 162-176.

[3] S. Nelakuditi, Z. L. Zhang, R. Tsang and D. Du, Adaptive Proportional Routing: a Localized QoS Routing Approach, IEEE/ACM Transactions on Networking 10 (2002), pp. 790-804.

[4] S. Alabbad, M. E. Woodward. "Localized Route Selection for Traffic with Bandwidth Guarantees", Simulation, Transactions of the SCS, Vol.83, No.3 (March 2007), pp. 259-272.

[5] A. Mohammad, M. E. Woodward. "Localized Quality Based QoS Routing", Proc. of International Symposium on Performance of Computer and Communications systems IEEE press, SPECTS June 2008

[6] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A Framework for QoS based Routing in the Internet," in *IETF RFC 2386*, 1998..

[7] G. Apostolopoulos, R. Guerin, S. Kamat and S. K. Tripathi, "Quality of Service Based Routing: A Performance Perspective", *ACM SIGCOMM Computer Communications Review*, vol. 28, issue 4, pp. 17-28, October 1998.

[8] A. Varga, The OMNeT++ Discrete Event Simulation System, the European Simulation Multiconference, Prague, Czech Republic, 2001.

[9] S. Chen, an K. Nahrstedt, "Distributed Quality of Service routing in High Speed Networks Based on Selective Propping, 23rd Annual Conference on Local Area Networks (LCN'98), IEEE, 1998, pp. 80-89.

[10] A. Medina, A. Lakhina, I. Matta and J. Bayers, Brite: An approach to universal topology generation, proceedings of the International Workshop on Modeling Analysis and Simulation of Computer and Telecommunications Systems, Cincinnati, Ohio, 2001, pp. 346-356.

[11] B. M. Waxman "Routing of Multipoint Connections", IEEE Journal on Selected Areas on Communications vol.6, No.9, pp 1617-1622, 1998.

[12] AS. Shaikh, J. Rexford and K. Shin, Efficient Precomputation of Quality of Service Routes, International Workshop on Network and OS support of Digital Audio and Video (NOSSDAV'98), IEEE, Cambridge, 1998, pp. 15-27.

[13] A. Mohammad, M. E. Woodward. "Effects of Topology on the Performance of Localized QoS Routing Algorithms", Proc. of International Symposium on Performance of Computer and Communications systems IEEE press, SPECTS July 2009.

Abdulbaset Mohammad is a Ph.D candidate and a member of networks and performance engineering research group at the University of Bradford with an expected completion in December 2010. He received the B.Sc degree in Communication Engineering from Libya in 1990 and M.Sc degree from Technical University of Budapest in Telecommunication Engineering in 2000 with excellent honor thesis in the field of routing algorithms in ATM networks. He is working as a lecturer in Libya's higher education. His current research interest includes QoS routing algorithms, call admission control algorithms, complex networks and topology based routing algorithms. He is a member of the IEEE and a member of the SCS.