Localised Routing Algorithms in Communication Networks with Quality of Service Constraints. Performance Evaluation and Enhancement of New Localised Routing Approaches to Provide Quality of Service for Computer and Communication Networks.

Item Type	Thesis
Authors	Mohammad, Abdulbaset H. T.
Rights	 The University of Bradford theses are licenced under a Creative Commons Licence.
Download date	2025-04-19 21:32:16
Link to Item	http://hdl.handle.net/10454/5163

University of Bradford eThesis

LOCALISED ROUTING ALGORITHMS IN COMMUNICATION NETWORKS WITH QUALITY OF SERVICE CONSTRAINTS

Performance Evaluation and Enhancement of New Localised Routing Approaches to Provide Quality of Service for Computer and Communication Networks

by
Abdulbaset H. T. Mohammad

PhD

School of Computing, Informatics and Media

University of Bradford

December 2010

Abdulbaset Mohammad confirms that the work presented in this thesis is my investigation. Any information has been presented from other sources I confirm that has been indicated in the thesis.

School of Computing, Informatics and Media

University of Bradford

Abstract

Keywords: Localised quality of service routing, performance evaluation and routing algorithms.

The Quality of Service (QoS) is a profound concept which is gaining increasing attention in the Internet industry. Best-effort applications are now no longer acceptable in certain situations needing high bandwidth provisioning, low loss and streaming of multimedia applications. New emerging multimedia applications are requiring new levels of quality of services beyond those supported by best-effort networks.

Quality of service routing is an essential part in any QoS architecture in communication networks. QoS routing aims to select a path among the many possible choices that has sufficient resources to accommodate the QoS requirements. QoS routing can significantly improve the network performance due to its awareness of the network QoS state.

Most QoS routing algorithms require maintenance of the global network's state information to make routing decisions. Global state information needs to be periodically exchanged among routers since the efficiency of a routing algorithm depends on link-state information accuracy. However, most QoS routing algorithms suffer from scalability due to the high communication overhead and the high computation effort associated with maintaining accurate link state information and distributing global state information to each node in the network.

The ultimate goal of this thesis is to contribute towards enhancing the scalability of QoS routing algorithms. Towards this goal, the thesis is focused on Localised QoS routing algorithms proposed to overcome the problems of using global network state information. Using such an approach, the source node makes routing decisions based on the local state information for each node in the path.

Localised QoS routing algorithms avoid the problems associated in the global network state, like high communication and processing overheads. In Localised QoS routing algorithms each source node maintains a predetermined set of candidate paths for each destination and avoids the problems associated with the

maintenance of a global network state by using locally collected flow statistics and flow blocking probabilities.

بسم الله الرحمن الرحيم

In the name of God, most merciful, most beneficent

Acknowledgments

First and foremost, I thank Allah, the Almighty God for his unlimited grace and for giving me blessing to complete this thesis. Pursuing a PhD study is a very rewarding experience. It is a stressful process that one can not go through alone. However, this thesis marks a milestone that would have been unreachable without the care and support of many individuals.

I wish to express my deepest thanks to my supervisors: Professor Michael E. Woodward for his direct guidance and continuing support throughout the period of my graduate study. Professor Irfan Awan for his help, despite all the challenges I faced during my PhD journey, I would like to give him special thanks for his administrative support. Their guidance has armed me with skills that I will use to develop my academic career.

I was also very lucky to work with the NetPEn Research Group of very supportive colleagues.

Any success I have attained during my academic career I attribute to my parents Al-Hadi Al-Tuhami and Halima Al-Thaib, who encouraged me in every endeavour and taught me to value education. They have been a great emotional support throughout my life. Undiminished love and deepest gratitude is devoted to my wife Ibtsam Ezzeden who stood by me selflessly as I pursued my goals. Having my wholesome children, Yahya, Abdulmoeiz and Malik by my side made this long journey full of joy. My special gratitude is due to our family for their passionate support and patience.

Finally, I would like to thank Libya’s higher education for granting me the scholarship and for providing full financial support during the period of my graduate studies.

List of Publications

[1] A. H. Mohammad. "A new Localised Network based Routing Model in Computer and Communication Networks", International Journal of Computer Networks & Communications, Vol.3, No.2 pp. 146-163, March 2011.

[2] A. H. Mohammad. "New Localised Call Admission Control Algorithms in Communication Networks with Quality of Service Constraints", International Journal of Computer Science and Network Security, Vol.10, No.11, pp. 125-131, November 2010.

[3] A. H. Mohammad, M. E. Woodward. "Localised QoS Routing and Admission Control for Congestion Avoidance", *Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems*, IEEE Press, CISIS February 2010.

[4] A. H. Mohammad, M. E. Woodward. "Localised QoS Routing and Admission Control: Performance Evaluation", *Proceedings of the 6th International Conference on Innovations on Information Technology*, IEEE Press, December 2009.

[5] A. H. Mohammad, M. E. Woodward. "Effects of Topology on the Performance of Localised QoS Routing Algorithms", *Proceedings of the International Symposium on Performance of Computer and Communications Systems*, IEEE Press, SPECTS July 2009.

[6] A. H. Mohammad, M. E. Woodward. “Localised Quality Based QoS Routing”, *Proceedings of the International Symposium on Performance of Computer and Communications Systems,* IEEE Press, SPECTS, June 2008.

Table of Contents

List of Figures

List of Tables

List of Principal Abbreviations

ABR	Acceptance Rate Based Routing
ATM	Asynchronous Transfer Mode
AS	Autonomous System
BA	Barabási and Albert Model
BCM	Balance Clustering Metric
CAR	Congestion Avoidance Routing
CBR	Constraint Based Routing
CBR	Credit Based Routing
DRA	Distributed Routing Algorithm
DiffServ	Differentiated Services
ER	Erdős and Rényi Model
FTP	File Transfer Protocol
IETF	Internet Engineering Task Force

IntServ	Integrated Services
IP	Internet Protocol
ISIS	Intermediate System to Intermediate System
ISP	Internet Service Provider
ITU-T	International Telecommunication Union - Telecommunications
LSA	Link State Advertisement
MPLS	Multi-Protocol Label Switching
NED	Network Description Language
OMNET++	Objective Modular Network Testbed in C++
OSPF	Open Shortest Path First
PNNI	Private Network to Network Interface
PSR	Proportional Sticky Routing
QBR	Quality Based Routing
QoS	Quality of Service

QOSPF	QoS extensions to OSPF Protocol
RIP	Routing Information Protocol
SP	Shortest Path
SRA	Source Routing Algorithm
RVSP	Resource Reservation
SDP	Shortest Distance Path
SWP	Shortest Widest Path
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
TE	Traffic Engineering
VoIP	Voice over IP
WSP	Widest Shortest Path
WWW	World Wide Web

List of Principal Symbols

$\widehat{A}$	Adjacency matrix
b	Requested bandwidth
$bw(l)$	Available bandwidth for a link l
C	The maximum capacity of the links in the network
$\bar{C}$	Clustering coefficient
D	Diameter of the topology
d_u	Degree of the node u
$\langle d \rangle$	The average node degree
$G(V,L)$	A network G with nodes V and links L
$\bar{h}$	The average path length in the network
L	The number of links in the network
M	Sliding window length
$\widehat{M}$	Path matrix
$\bar{m}$	The arithmetic mean
N	The number of nodes in the network
$O(N)$	The complexity of order N
P	Path

$P\left\langle d\right\rangle$	The degree distribution
P^{alt}	The alternative path with maximum credit
$P^{\min}$	The minimum hop path with maximum credit
$P^{\min}.Credits$	The credit of the minimum hop path for CBR
$P^{alt}.Credits$	The credit of the alternative path for CBR
$P.Quality$	The average path quality of the path for QBR
$P.Rate$	The average acceptance rate of the path for ABR
$P.Avg$	The average bandwidth of the path for CAR
R	The candidate path set
$R^{\min}$	The set of minimum hop candidate paths
R^{alt}	The set of alternative candidate paths
T	The sum of off-diagonal elements in the path matrix
v_{ij}	The distance between node i and node j
Λ_i	The sum of row distances in the path matrix

Chapter 1

Introduction

1.1 Background

Recent years have witnessed an exponential growth of the global Internet, both in its size and subscriber population, as well as in the variety of ways it is used. The traffic volume of the Internet continues to grow exponentially due to the emergence of new characteristics and new requirements of real-time traffic applications on the Internet.

Despite the fact that the Internet has witnessed tremendous success, its service model and architecture have remained the same. In packet switched networks, such as the Internet, routing is a process of delivering packets from sources to their ultimate destination via the most appropriate path. Routing in the Internet is still a datagram routing network; it was originally designed to provide a “best-effort” paradigm in which all packets would be served indistinguishably. The best-effort traffic is processed as quickly as possible. But there is no guarantee as to timeliness or actual delivery. The network itself does not actively differentiate between applications over the network. All packets are dynamically treated in the same fashion. The best-effort services are well suited for traditional data applications such as telnet, e-mail, and file transfer (ftp), they are inadequate for many emerging applications such as IP telephony, video on demand and teleconferencing, which require bandwidth guarantees and stringent delay.

In order to address this issues, multi-path traffic techniques have been proposed [1], whereby Quality of Service (QoS) based routing is an essential part in any QoS architecture. The function of QoS routing is to select a path, among the many possible choices, that has sufficient resources to accommodate the QoS requirements. QoS routing can significantly improve the network performance due to its awareness of the network QoS state. This thesis mainly addresses the issues and problems of QoS routing.

1.2 Motivation

New emerging multimedia applications are appearing over the Internet, demanding particular QoS requirements, such as bandwidth, which must be taken into account when selecting a path.

A key aspect in these QoS architectures is the routing process, i.e. how routes are computed, selected and established. In packet switched networks, routing can be defined as the act of moving information across a network from a source to a destination. Therefore, QoS routing can be defined as moving information across a network from a source to a destination, while considering QoS requirements in order to achieve more satisfaction and greater optimisation of network resource usage.

It is worth noticing that there are two different tasks when talking about routing: the first task is collecting routing information to collect state information about the topology and link states of the network, the second task involves a routing algorithm to make a routing decision which will find a feasible path for a new connection based on the collected information. There are inherent scalability issues for both tasks. As for the first one, collecting the global network QoS state gives rise to several problems:

- The global network maintained at every node has to be updated frequently enough to cope with the dynamics of network parameters, such as bandwidth and delay, which imposes a large communication overhead for large scale networks.
- To keep absolutely up-to-date information would require a prohibitively extensive exchange of link state update information between network

nodes for all links and this would consume unacceptable amounts of network resources.

- The link state algorithm can only provide an approximate global state due to the overhead and non-negligible propagation delay of state messages, this imprecise aspect of the global state has a drastic effect on the resulting performance [2].
- Out-of-date information due to large update intervals can cause route flapping [2] in global QoS routing algorithms, which results from global synchronisation of distributing the network state, such behaviour results in poor route selection, instability and an overall degradation of network performance.
- As for the second task, the computation overhead at the source is excessively high due to the frequent updates needed, which for QoS routing is typically done on a per-connection basis.

The above problems become worse as the size of the network increases and so the scalability of QoS routing algorithms becomes a major challenge [20]. The inherent scalability problems associated with global view approaches have created a promising challenge to look for alternative approaches that can cope with the complexity and the dynamics of large-scale networks. These challenges and inherent scalability problems of global QoS routing algorithms therefore motivate or agreed to investigate new methods and techniques for QoS routing algorithms in the context of the tasks described above. For this thesis, it is motivated to generate impetus in research and develop QoS routing algorithms using the local state information by stimulating QoS routing designers to think "*out-of-the-box*" by focusing on the development of novel QoS routing models incorporating holistic aspects of QoS performance objectives, the Internet topological characteristics, and infusion of other concepts and ideas related to the area of research.

1.3 Thesis Aims and Objectives

The research in this thesis is focused on contributing towards enhancing the inherent scalability of QoS routing algorithms. These aims can be divided into the following sub aims, as listed below:

- To develop scalable Localised QoS routing algorithms that select a path based on available resources.
- To develop techniques that can be used to design using characteristics of the suitable topology models in order to enhance their scalability.

These aims can be achieved through the following objectives:

- To study all aspects of global and local QoS routing focusing on their related scalability problems.

- To focus on source routing algorithms and study attributes that make them scalable in relation to various dependencies on the state information and QoS routing problems.
- To develop a simulation environment that will be used to assess and compare the performance of various QoS routing algorithms.
- To develop Localised QoS routing algorithms that provide guaranteed QoS requirements, low message overhead and efficient resource utilisation.
- To develop an efficient and scalable Localised QoS routing algorithm that uses acceptance rate path selection methods.
- To develop an efficient and scalable Localised QoS routing algorithm that uses a bandwidth, the same QoS metric used as a path selection metric.
- To develop a network model that can be used to predict and enhance the performance of Localised QoS routing algorithms.
- To develop an efficient and scalable Localised QoS routing algorithm that avoids congestion in order to achieve balanced traffic distribution.
- To develop an efficient and scalable Localised QoS routing algorithm that uses a Localised QoS routing algorithm that is supplemented by an admission control in order to achieve efficient utilisation of network resources.

1.4 Thesis Original Contributions

Throughout the thesis we make the following contributions and achievements [118] [119] [120] [121] [122] [125]:

- The Localised Acceptance rate path selection Based Routing algorithm (ABR) is developed for QoS routing that performs routing using only flow statistics collected locally, which uses an average path statistics.
- A new Localised Quality Based QoS Routing (QBR) algorithm is developed using bandwidth (the same QoS metric used) as a new path selection method. To the best of our knowledge using bandwidth as a path selection criterion is the first in the context of Localised routing algorithms. The performance is compared against an existing Localised Credit Based Routing (CBR) algorithm and demonstrates through extensive simulations that the QBR algorithm performs well in comparison with the CBR algorithm [121].
- A new topology parameter is developed to increase the flexibility of a large number of possible candidate paths that can be used to predict which network topologies gives better performance on the quality of Localised routing algorithms, taking path length into consideration. To the best of our knowledge this study is the first in the context of routing algorithms and topology based models [120].

- A new Balanced Clustering Metric model (BCM network) is developed using the BCM metric that can be rewired to introduce increasing the connections that the network accepts.
- The Localised Congestion Avoidance Routing (CAR) algorithm is developed in which the decision of routing is taken based on information updated in each connection request which distributes the load efficiently. CAR algorithm made transition from static update information to dynamic update information. The performance is compared against existing Localised CBR and QBR routing algorithms and demonstrate through extensive simulations that the CAR algorithm performs well in comparison with these [118].
- A new Call Admission Control with Localised QoS routing algorithm (CAC) is developed in order to limit QoS routing decisions to edge routers and achieve an efficient utilisation of network resources. To the best of our knowledge the call admission control has not been studied in the context of Localised routing algorithms. Simulations of various topologies are used to illustrate the integrated of Localised routing with admission control algorithm [119] [122].

1.5 The Thesis Outline

The structure of this thesis is organised as follows:

Chapter 2 introduces the QoS routing concepts and algorithms, provides a description of QoS routing notation and metrics, and considers how state information is maintained and update frequency is provided. An explanation about the QoS-routing problem is given to discuss the advantages and limitations of different routing strategies. This chapter also provides an overview of QoS routing algorithms based on shortest path algorithms, global routing algorithms and Localised routing algorithms.

Chapter 3 gives a description of the simulation model used to assess the performance of the algorithms proposed. It also describes the simulator design and how it was validated. Different types of network topologies, parameter settings and performance metrics used in the performance evaluation are also described.

Chapter 4 presents the ABR algorithm and a Localised QBR routing algorithm that relies on using bandwidth (the same QoS used) and evaluates their performance.

Chapter 5 looks at the varying performance of any routing algorithm. Since it is highly dependent on the underlying network topology, this chapter shows development of a new metric that can be used to predict which network topologies give better performance. A new topology parameter (BCM metric) to increase the flexibility of a large number of possible candidate paths that can be used to predict which network topologies gives better performance on the quality of Localised routing algorithms.

Chapter 6, the Localised call admission control algorithm is proposed to utilize the network resources efficiently and reduce the signalling overhead.

This thesis is concluded in **Chapter 7** by presenting related conclusions of the proposed algorithms and providing some key recommendations for future directions for related work.

Chapter 2

Quality of Service Routing: Concepts and Algorithms

2.1 The Need for Quality of Service

Packet switched networks, such as the Internet, are a collection of networks interconnected to each other through gateway routers. Routing is primarily defined as the process of delivering flows from sources to their ultimate destinations through intermediate nodes called routers, via the appropriate path.

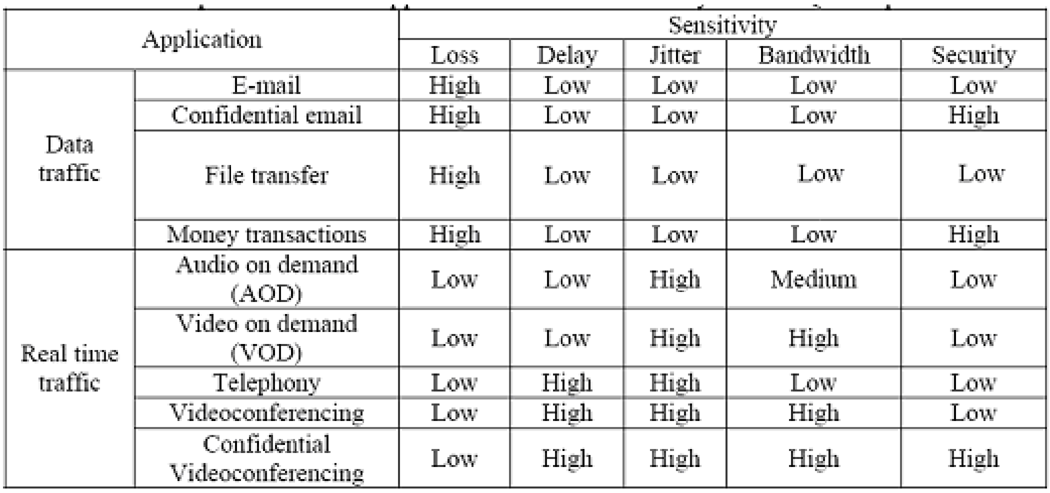
The routing process consists of two main functions: the first function entails selecting a path between a source and a destination and the second entails forwarding messages to their destination upon the path selected. The first function is carried out by using routing algorithms, such as Dijkstra's and Bellman-Ford's shortest path algorithms [3] [4]. The second function involves routing protocols, such as Open Shortest Path First (OSPF) [5] and Routing Internet Protocol (RIP)

[6] which are responsible for forwarding, and are usually directed by locally-stored databases within routers, which maintain a record of the best routes [7].

These databases are called routing tables and the construction of such routing tables is usually used to find efficient routing. Packet switched networks can be classified into two types based on the times at which routing decisions are made [5]. The first type is the datagram network, where two packets of the same user pair may travel along different paths, and a routing decision is necessary for each individual packet.

The Internet today, via the Internet protocol (IP) only provides single level best-effort applications where the data packets of all applications are judged equally without any guarantees on the service qualities and timelines. Routing in the current Internet focuses primarily on connectivity and typically supports only the best-effort datagram service. The routing protocols deployed, such as OSPF supports only the best-effort service. While these protocols are sufficient for traditional applications such as file transfer protocol (ftp) and telnet, they are inadequate for many emerging applications such as IP telephony (i.e. VoIP), video on demand and teleconferencing, which have demanded stringent service assurance of high bandwidth, explicit delay bounds, stable jitter and low packet loss, which constitutes the notion of "Quality-of-Service" (QoS). Table 2.1 lists several common applications and the stringency of their QoS requirements.

Moreover, in best-effort networks, there is no admission control or resource reservation for each connection because best-effort networks do not maintain information about each connection in routers along the path.



Application		Sensitivity				
		Loss	Delay	Jitter	Bandwidth	Security
Data traffic	E-mail	High	Low	Low	Low	Low
	Confidential email	High	Low	Low	Low	High
	File transfer	High	Low	Low	Low	Low
	Money transactions	High	Low	Low	Low	High
Real time traffic	Audio on demand (AOD)	Low	Low	High	Medium	Low
	Video on demand (VOD)	Low	Low	High	High	Low
	Telephony	Low	High	High	Low	Low
	Videoconferencing	Low	High	High	High	Low
	Confidential Videoconferencing	Low	High	High	High	High

Table 2.1 Applications and the stringency of their QoS requirements

In addition, due to packets being routed independently, they may take different paths and arrive at the destination out of order. There is no guarantee of packet delivery additional functions in the host that are required to provide reliable transmission for certain applications, such as file transfer. The Transmission Control Protocol (TCP) [7], on the end systems, provides reliable packet delivery by retransmission after a failed delivery which is detected by the lack of acknowledgement or a timeout. Other applications that do not require more than a best-effort service can directly use this type of network. The User Datagram Protocol (UDP) [8] serves the need of these types of applications.

However, routing protocols in best-effort networks, such as the Internet have several drawbacks. Firstly, they primarily use a shortest path routing only paradigm that uses only a single minimum hop path metric between source and destination pairs, which can cause unbalanced traffic distribution. They also use single routing metrics, such as hop count.

Since there is no admission control and they do not differentiate between traffic that uses the network, all connection requests are accepted to the network and there is no connection rejection. Clearly, when the offered load exceeds available network capacity, network congestion will occur. Therefore, the problem of congestion is inherent in the best-effort network because it lacks a traffic control mechanism.

The complete definition of QoS is given as follows:

- QoS is a set of service requirements to be met by the network while transporting a flow [10].

- QoS is the collective effect of service performances which determine the degree of satisfaction of a user of the service (ITU-T recommendation E-800) [11].

2.2 Internet Quality of Service: Architectures and Mechanisms

The Internet Engineering Task Force (IETF) has introduced many mechanisms and service architectures for quality of service provisioning on the Internet. Most noted is the Integrated Services/Resource Reservation (IntServ/RVSP) Signalling Protocol architecture [12], [13], the Differentiated Services (DiffServ) architecture [14], Multi-Protocol Label Switching (MPLS) [15], Traffic Engineering (TE) [1] and Constraint Based Routing (CBR) [7].

In Integrated Services (IntServ), the QoS requirements of a given flow can be guaranteed by reserving network resources, such as bandwidth and buffers, defined by the flow's requirements at the source and the destination prior to data transmission. IntServ relies on the RVSP [16] to reserve network resources for an application's traffic flow. It is a per-flow strategy, does not scale well topologically, is wasteful of resources, especially bandwidth, and increases complexity on end-systems. This is because the state of each flow must be recorded and maintained in the intermediate routers on the delivery path for the active duration of each flow to provide QoS requested [16].

Differentiated Services (DiffServ) define per-hop packet forwarding mechanisms that could differentiate classes of flows at each link. DiffServ have been developed to solve the scalability problem raised by Integrated Services. A network service provider can define service classes and allocate appropriate network resources by configuring the routers. Routers in integrated services differentiate between large numbers of connection requests, which result in large overheads to maintain a connection request state. DiffServ does not require advanced path information and flow setup making it scalable and flexible. In IntServ, the network tunes its resources to an application's flow, while in DiffServ a user tunes an application's flow to the network services and resources awarded.

Although DiffServ is more scalable and has fewer signalling overheads than IntServ/RVSP, it does not provide end-to-end guarantees for real time applications and does not provide guarantees during congestion.

Multi Protocol Label Switching (MPLS) is a new packet forwarding scheme where short, fixed length labels are used to differentiate different service classes [1]. MPLS has a header between the network layer and data link layer in the IP header [17]. MPLS can set up an explicit path using a label switched in the MPLS domain, since the path used by the packet is specified by a single label switch router and should not be carried in the packet header.

MPLS delivers additional capabilities for providing quality of service and engineering network traffic. By decoupling the forwarding mechanisms from routing, MPLS enables the deployment of specialised routing services without changes to the packet forwarding path [18]. In particular, MPLS supports setting up for explicitly routed label switching protocols in which the entire path is determined at the source. This allows for rerouting of certain classes of traffic to balance network load independent of the underlying conventional shortest-path routing protocol [1].

Traffic Engineering is the process of arranging how traffic flows through the network so that congestion caused by unbalanced network utilisation can be avoided. Unbalanced traffic distribution is mainly caused by current routing protocols [3] such as Routing Information Protocol (RIP), Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (ISIS) that always select the shortest path to forward packets. The network traffic engineer identifies classes of traffic and their endpoints, along with associated QoS metrics and policies.

Perhaps the most important QoS provisioning mechanism perceived is Constraint Based Routing (CBR) [19]. The goals of CBR are to compute paths that are subjected to multiple constraints and to increase utilisation of the network [19]. The MPLS mechanism also specifies a CBR that enables demand-driven routing capable of satisfying resource and policy requirements. A constraint based routing protocol then aims to find suitable paths that meet the requirements. Constraint based routing operates much the same as source-directed link-state QoS routing protocols. The value of constraint-based routing is in the fact that it has the ability to assist in traffic engineering while reducing the amount of manual route configuration [7].

Constraint based routing also considers network configuration, flow requirements and resource availability on links. The constraints of CBR can either be network administrative policies for using the network resources '*policy routing*' or simply the QoS requirements of a flow '*QoS-routing*' such as bandwidth, delay, jitter and loss [20]. The absence of QoS-routing renders IP quality of service architectures of IntServ and DiffServ to be dysfunctional, because neither have in-built path computation mechanisms. IntServ and DiffServ must depend on QoS-routing in order to compute feasible paths that would have the required network resources to accept the flows. QoS-routing is complementary to traffic engineering making the process automatic [21]. The QoS-routing position in the ITU-T framework [22] is depicted in Figure 2.1.

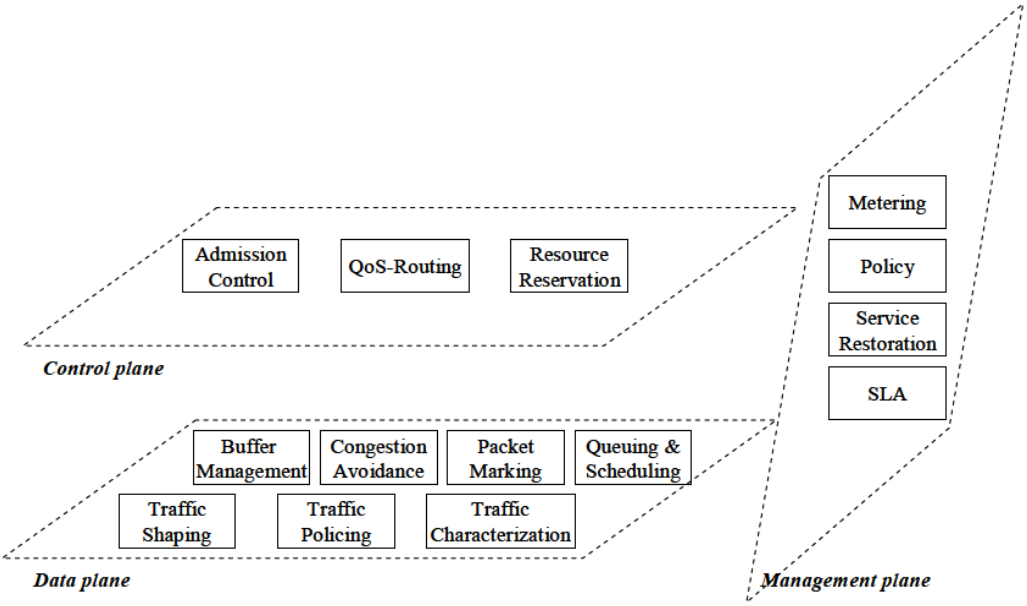


Figure 2.1 The QoS-routing position in the ITU-T QoS architectural framework

2.3 Quality of Service Routing Paradigm

QoS routing is an essential part in the QoS architecture. Before any connections can be made or any network resources reserved, a feasible path between a source-destination pair must be established.

QoS-routing is a routing mechanism under which paths for flows are determined based on some knowledge of resource availability in the network as well as the QoS requirements of flows [10].

Flow requirements for QoS-routing are frequently referred to as the metrics of QoS-routing, such as bandwidth, delay, and number of hops, cost and jitter.

The fundamental objectives of QoS-routing are threefold:

1. Dynamic determination of feasible paths that have sufficient available resources capable of satisfying the QoS requirements.
2. Network resource optimisation and improvement of overall performance by efficient distribution of the traffic in the network and maximisation of its network resources.
3. Adaptability regarding network congestion, which provides smooth performance degradation to lower priority traffic.

2.3.1 Notations and Metrics

A network topology can be represented as a directed graph $G(V, L)$, where V represents the number of nodes and L represents the number of links that connect nodes in the network.

Each directed link belongs to L from node *u* to node *v*, represented by *(u, v),* and has *k* values of its QoS metric, one for each directed link, represented as *k* weights: $w_1(u,v), w_2(u,v), w_3(u,v),...., w_k(u,v)$ where $w_i(u,v) > 0$ and $\forall (u,v) \in L$. Path P from node *S* to node D in *G* is represented by $P = u_{S=0} \rightarrow u_1 \rightarrow u_2 \rightarrow ... \rightarrow u_{D=j}$ such that $(u_i, u_{i+1}) \in L$ for all values of i where $1 \leq i \leq j-1$.

In order to find the path *P* that satisfies QoS requirements, a flow is needed by QoS routing algorithms. These QoS requirements can be presented as a single metric or a combination of different metrics. The three most common types of QoS metrics are also referred to as the composition rules of the metrics, as follows [23]:

- The QoS metric is **concave** if: $w(P) = \min(w(u1, u2), w(u2, u3),........, w(ui\text{-}1, ui))$. Bandwidth is the most widely used metric, this is a concave metric, it is also called a link residual bandwidth metric.
- The QoS metric is **additive** if: $w(P) = w(u_1, u_2) + w(u_2, u_3) + ... + w(u_{i-1}, u_i)$. Delay, link cost and hop count are additive metrics.
- The QoS metric is **multiplicative** if: $w(P) = w(u_1, u_2).\ w(u_2, u_3).........\ w(u_{i-1}, u_i)$. Loss rate and reliability are multiplicative metric.

Pruning can be used with non-additive metrics to reduce computation overhead by removing the links that do not satisfy QoS requirements. Multiplicative metrics can be considered as additive metrics by replacing the link weights and constraint by their logarithms.

2.3.2 The State Information Structure

The routing process essentially consists of two basic tasks. The first task is to collect the state information and keep this information up to date. Each node in the network provides a consistent view of network state information. The procedure followed to perform the first task is sometimes referred to as the '*routing protocol*'. The second task is to find a feasible path for a new connection based on the information collected in the first task. This task is carried out by a so-called '*routing algorithm*'. The performance efficiency of any routing algorithm directly depends on solving of the first task. The state information structure takes the form of global, local and aggregated information, as described below:

- **Local state information**

Network nodes (i.e. routers) gather recent or the latest state of its local traffic information statistics and assume to maintain its up-to-date local state. This can be the residual link bandwidths or any other QoS metric. The state information collected is then used to find a feasible path for a new connection [24].

- **Global state information**

The consolidation of local state information of all nodes in the network into a single database is called global state information. Every node is able to maintain the global state information by either a link state protocol or a distance-vector protocol [20], which exchanges the local states among the nodes periodically.

There are two protocols whereby collection and maintenance of the global state is carried by each node [3]: link state protocol and distance vector protocol, which exchange the local states among the nodes periodically. The distance vector protocol is a simple routing protocol that uses distance metrics among adjacent nodes to choose the best path. In distance vector protocol each node has a routing table (vector) giving the best-known distance (according to the metric) to each destination. These routing tables are updated by exchanging information with neighbours and it is assumed that each node knows the distance to each of its neighbours.

The link state protocol updates other nodes in the network in order to determine the best path. Each router discovers its neighbours, measures the required QoS metrics and sends information to all other nodes in the network, this enables each node to see a topological view and the state of every link (local states). When the network size increases the global state becomes imprecise due to an approximation of the current state caused by propagation delay of local states [20].

- **Aggregated state information**

In large networks, maintaining the global-state database is impractical as it is wasteful of resources due to excessive local state information which causes severe information imprecision. An approach proposed to make path computation scalable is aggregation of the state information. Aggregation is done by making the network into a hierarchical structure [25], [20], [26]. Nodes are grouped into domains at multiple layers forming a multi-domain hierarchy of nodes.

The multi-domain hierarchy of nodes is virtual and does not alter the physical connection of nodes in the network. Each domain is represented by a logical node. Nodes in each domain keep a detailed global state of the domain maintaining an aggregated state of other domains. In spite of being able to achieve scalability and reduction in local state information, state aggregation still adds additional imprecision [27].

2.3.3 State Information Inaccuracy

The performance of QoS routing schemes rely on the maintenance of the global state, in general there is a tradeoff between the overhead of frequent updates and the accuracy of global state information that the path selection algorithm depends on [2]. Generally, providing up-to-date changes of all links at all times are impractical, as this would require a huge amount of resources consumed by rapid global state updates in each link state. In the OSPF protocol it is recommended that the link state is updated once every 30 minutes. However, updating at such a long interval will result in some link state changes not being advertised [28].

In order to reduce the protocol overhead a limited number of updates is required, but the large time interval leads to a stale link state, which may affect QoS routing in different ways, as follows [29]:

- The routing algorithm may hide the fact that there is a feasible path in existence.
- A path may be rejected in the setup process because the required capacity cannot be provided for all links.
- A non-optimal path may be selected by the QoS routing algorithm while more suitable paths exist.

Maintenance of accurate network state information usually requires an insignificant amount of routing updates, since network state information changes are infrequent. However, maintaining accurate QoS state information requires a much greater amount of link state updates because network resource availability, such as available bandwidth, changes with each flow arrival and departure.

2.3.4 QoS Routing Problems

The focus is on studying routing problems which involve finding a path that satisfies a set of QoS constraints between source and destination pairs using the unicast routing problem. The QoS requirements of a connection are given as a set of QoS constraints, and they can be characterised into '*link-constraints*' and '*path-constraints*'.

- A link-constraint imposes a maximum or minimum limit on the use of link resources. Bandwidth and buffer space are examples of link-constraints.

- A path-constraint imposes a limit on the end-to-end QoS on a path. Delay, cost, jitter and hop count are examples of path-constraints.

	Problem	Example	Description	Complexity
Basic routing problems	Link-constrained	Bandwidth- constrained	Find a path whose bottleneck bandwidth is above a given value	polynomial
	Link-optimization	Bandwidth- optimization	Find the path with maximum bottleneck bandwidth	
	Path-constrained	Delay-constrained	Find a path with bounded delay	
	Path-optimization	Cost-optimization	Find a path whose total cost is minimized	
Composite routing problem	Link-constrained Path-optimization	Delay- optimization Bandwidth- constrained	Find the least-delay path with the required bandwidth	polynomial
	Path-constrained Link-optimization	Delay- constrained Bandwidth- optimization	Find the path with maximum bottleneck bandwidth and bounded delay	
	Multi-path constrained	Delay-constrained Cost-constrained	Find a path whose cost and delay are less than some given values	NP- complete
	Path-constrained Path-optimization	Delay-constrained Cost-optimization	Find the least-cost path with bounded delay	

Table 2.2 Unicast QoS routing problems

Table 2.2 lists problems of unicast QoS routing problems. A routing problem is defined as finding a path that satisfies a set of QoS constraints from a source node to a destination pair (i.e. a unicast QoS- routing problem). Single constraint routing problems can be further divided into four different types.

They are:

1. *Link-optimisation* – Link-optimisation routing problems entail finding a path with the largest bottleneck bandwidth (i.e. a *widest path* problem [23]). The routing problem can be solved using polynomial time algorithms, such as the modified Dijkstra's [30] (or Bellman-Ford [31]).
2. *Link-constrained* – Link-constrained routing problems can be easily reduced to link-optimisation problems. One example is finding a path whose link bandwidths are equal to or higher than a specified bandwidth requirement.
3. *Path-optimisation* – Path-optimisation routing problems consist of finding a path whose end-to-end cost is minimised. One example is least-cost routing.
4. *Path-constrained* – Path-constrained routing problems impose an additive bound on the end-to-end QoS of a path. A common example is to find a path whose end-to-end delay does not exceed a specified delay bound (i.e. delay-constrained routing).

Path-optimisation and path-constrained routing problems are also solvable using Dijkstra's (or Bellman-Ford) SP algorithm.

A single constraint routing problem exists when a connection specifies exactly a single QoS requirement. However, multiple constraint routing problems exists when a connection specifies multiple QoS requirements.

A multiple constraint routing problem involving two or more additive metrics has been shown to be NP-Complete [32] [13] [15]. The metrics were assumed to take real values and be independent. This problem has been investigated and many heuristics have been proposed to solve the NP-complete problem in polynomial time [1] [7] [33] [34]. The problem can be avoided if one of the metrics refers to bandwidth as the QoS [33].

2.3.5 QoS Routing Strategies

The main tasks of QoS routing are collecting state information, keeping the state information up to date, and searching the routing for the information to find feasible paths for flows. Various QoS routing algorithms are classified into three types based on the way the state information is maintained and the search for a feasible path is carried o

2.3.5.1 Source Routing

In the source routing algorithms [20] [23] [35] [36] each node maintains the complete global state of the network. The process of finding a feasible path is carried out at the source node using the global state.

If there is no feasible path, the source node can immediately reject the connection or negotiate for less restricted requirements with the initiator of the connection. If the source node is able to find a feasible path, it sends a control message along the selected path to inform the intermediate nodes about the path and each node takes a number of well defined actions to setup the connection along the selected path.

Strengths and Weaknesses of Source Routing

Source routing achieves its simplicity by transforming a distributed problem into a centralised one. By maintaining a complete global state, the source node calculates the entire path locally. It avoids dealing with the distributed termination problem.

Source routing is used in today's Internet, for example, to force an alternative link for certain traffic flow, e.g. to avoid congestion. In an MPLS framework, source routing could be more practical as the labels would represent the explicit paths. It guarantees loop free routes. In addition, it is much easier to design centralised heuristics for some NP-complete routing problems than to design distributed ones. However, source routing has several problems.

- The global state maintained at every node has to be updated frequently enough to cope with the dynamics of network parameters, such as bandwidth and delay, which imposes communication overheads for large-scale networks.
- The link state algorithm can only provide an approximate global state due to the overhead concern and non-negligible propagation delay of state messages, this impreciseness in the global state has a drastic effect on resulting performance [20].
- Out-of-date information, due to large update intervals, can cause route flapping in global QoS routing algorithms, which results from global synchronisation of distributing the network state [2] [36].
- The computation overhead at the source is excessively high, which with regard to the QoS routing is typically done on a per-connection basis.

In summary, source routing has scalability problems. It is impractical for any single node to have access to detailed state information about all nodes and all links in a large network [36].

2.3.5.2 Distributed Routing

In distributed routing the path computation is distributed among the nodes. Control messages are exchanged among the nodes and the state information kept at each node is collectively used for the path search. Most distributed algorithms need to maintain a global state of the network in each node, while the routing is done on a hop-by-hop basis. Some distributed algorithms, such as flooding-based algorithms, do not require global state information to be kept at each node; instead, the hop-by-hop routing is done based on the local states.

Strengths and Weaknesses of Distributed Routing

The routing response time in distributed routing can be made shorter and the algorithm is more scalable than source routing. It is possible to search multiple paths in parallel for a feasible one to increase the chance of success.

Most existing distributed routing algorithms [20 [37] [38] require each node to maintain a global network state, while flooding based algorithms, which do not maintain a global state, have a higher communication overhead. The distributed routing algorithms share more or less the same problems as the source routing algorithms. It is also very difficult to design efficient distributed heuristics for NP-complete routing problems, because no detailed topology or link state information is available [38].

In addition, when the global states at different nodes are inconsistent, loops may occur. Loops generally make the routing fail because the distance vectors do not provide sufficient information for an alternative path.

2.3.5.3 Hierarchical Routing

Hierarchical routing has long been proposed to cope with the scalability problem of source routing in large inter-networks [39] [40] [41]. An aggregated global state described in Section 2.3.2 is used in hierarchical routing. In hierarchical routing, each node maintains partial global state information where groups of nodes are clustered into higher-level logical nodes.

Hierarchical routing algorithms directly use source routing algorithms to find feasible paths based on aggregated state information when connection requests arrive. Hierarchical routing also uses distributed routing through the path computation shared over many nodes.

Strengths and Weaknesses of Hierarchical Routing

Hierarchical routing retains many advantages of source routing and distributed routing. Hierarchical routing performs well in large inter-networks because the extent of state information maintained at each node is logarithmic to the size of the complete global state, which means that hierarchical routing scales well due to reducing the level of state exchange overhead, nodal storage and computational effort [42].

The major drawback of hierarchical routing is that the network state information is aggregated, and additional imprecision is introduced in hierarchical routing, which imposes a negative impact on the QoS routing algorithm [43].

2.4 QoS Routing Algorithms

The problems of QoS routing have been addressed in many contexts in the literature and various algorithms have been proposed for providing QoS routing. These algorithms differ in terms of the source or distributed path which is chosen, how global state information is exchanged or Localised state information observations and whether the widest or shortest path is selected. An extensive survey of various QoS routing algorithms can be found in the literature [20] [44] [59] [60]. For the purpose of this thesis it is possible to broadly categorize QoS routing into global QoS routing schemes that are based on global link state updates and Localised QoS routing schemes that are based on local path state observations. This thesis will focus on Localised QoS routing algorithms.

2.4.1 Global QoS Routing Algorithms

The majority of QoS routing algorithms proposed require a periodic exchange of link state information among network nodes in order to obtain a global view of the network QoS state information. Based on this current view of the network state information, a source node dynamically finds a feasible path for a flow originating from it and going to the destination. Such global routing algorithms primarily differ in path selection methods and how to exchange the global state information. Global routing algorithms have to deal with the trade-off between the resource usage and balancing network load, as well as between the accuracy of the network state and link state update frequencies. Detailed information about relationships between global routing algorithms and network conditions can be found in [46].

2.4.1.1 The Widest Shortest Path Algorithm

The Widest Shortest Path algorithm (WSP) [47] chooses the fewest number of hops among all feasible paths that satisfies the bandwidth constraints. If there are more than one feasible path with the same number of hops, then the one with the largest bottleneck bandwidth is selected. The WSP algorithm prefers shorter paths in order to minimize the usage of network resources. Links that do not satisfy flow requirements are eliminated using pruning, WSP is used based on assuming the pruned topology [48]. The performance differs between WSP with and without pruning. The WSP scheme prefers limiting resource consumption by giving the priority to the number of hops. The WSP scheme tries to choose lightly loaded paths in order to balance the load. In general, the WSP algorithm is the most studied in the literature and WSP results in better performance under heavy loads, while WSP also gives better performance under low loads and many variations can be derived by choosing different cost functions of the shortest path [49] [50] [51].

2.4.1.2 The shortest widest path algorithm

The Shortest Widest Path algorithm (SWP) [23] finds the largest bottleneck among all feasible paths. If there is more than one path with the widest feasible path existing, the one with the minimum number of hops is chosen. The SWP algorithm prefers the widest paths so that it can give higher priority to distributing network load efficiently and avoiding congestion on shorter paths.

2.4.1.3 The shortest distance path algorithm

The Shortest Distance Path algorithm (SDP) [52] [53] selects the most feasible path with the shortest distance. The link's distance is the inverse of the available bandwidth of that link. The overall distance of a path is the sum of distances over all the links along the path, according to the distance function:

$$disp(p) = \sum_{i \in p} \frac{1}{w(i)} \quad \text{.....................} \; (2.1)$$

where *w(i)* is the available bandwidth of link *i*.

The SDP algorithm prefers the least loaded paths and calculates these using number of hops [52]. It provides a method to balance by using the distance function. Furthermore, the SDP algorithm can be modified to solve the shortest cost using the following function:

$$disp(p, n) = \sum_{i \in p} \frac{1}{w(i)^n} \quad \text{.....................} \; (2.2)$$

where *w(i)* is the available bandwidth of link *i*.

Changing *n* in the SDP algorithm results in covering the spectrum between shortest path (*n*=0) and widest ($n \to \infty$) algorithms [53].

2.4.1.4 QoS extensions to the OSPF algorithm

The QoS extensions to the OSPF algorithm (QOSPF) [47] propose mechanisms to extend the OSPF routing protocol. The goal is to improve performance traffic with minimum changes to the existing OSPF protocol structure. This algorithm computes the number of hops needed to perform the path selection algorithm, but the available bandwidth is advertised at each link in the network.

It is assumed that the current link state information database needs to be presented at each node in the network topology. The number of hops and bandwidth are used in the path selection of the QOSPF algorithm, in which three variations of the widest shortest path (WSP) are introduced with different computation methods and storage requirements. The first algorithm calculates every source node to pre-compute the maximum bandwidth with the minimum number of hops for every possible destination. This is done by the Bellman-Ford algorithm which identifies the optimal path (with maximum bandwidth) between the source and all destinations among the paths.

The second algorithm performs on demand routing by using a standard Dijkstra shortest path to calculate a feasible path to the destination based on a graph, from which all links with insufficient bandwidth are pruned. This algorithm does not need any QoS routing table and uses current up-to-date link state information, however the first algorithm has lower computational requirements due to its pre-computation property.

The third algorithm pre-computes the minimum number of hops using a standard Dijkstra calculation to find paths for all possible destinations with a set of quantised bandwidth values. That is, the range of bandwidth capacity is mapped to a fixed number of classes in order to find the minimum number of hops for a given bandwidth. The price to be paid for this algorithm is a loss in the accuracy of the generated path due to quantisation.

Several recent proposals exist in the literature addressing global QoS routing information inaccuracy and its effectiveness on the path selection process [36] [87] [88] [89]. With a different approach, [91] seeks to eliminate the inaccuracy of the QoS routing scheme based on a centralised server, while [92] focuses on avoiding the traffic engineering information. Whereas [90] predicts the availability of resources along a path, while eliminating the typical link state advertisements. This thesis focuses on bandwidth constraint applications using localised QoS routing algorithms [55].

2.4.2 Localised QoS Routing Algorithms (Related Work)

Localised quality of service routing schemes [54] [55] have recently been proposed as a viable alternative to the QoS routing algorithms that use global state information, where routing decisions are based on the whole network state. They try to circumvent the problems associated with maintaining global state information and the stateless problems by having source nodes infer the network QoS state based on flow statistics collected *locally*.

The Localised view of the network QoS state is used to perform flow routing. In Localised QoS routing algorithms each source node maintains a predetermined set of candidate paths for each possible destination to send the flow along these paths. These should be selected using one of the existing methods or by finding out which mechanism gives the best results. Localised QoS routing algorithms avoid the problems associated with the maintenance of a global network state by using locally collected flow statistics and flow blocking probabilities.

Localised QoS routing also avoids the drawbacks of selecting a best-path routing approach, such as degraded performance, in the presence of inaccurate routing information. The Localised approach considers adjustments for selecting a path at a network node by dynamically judging its quality and traffic flow based on local state information. Furthermore, the Localised routing approach has several important advantages: increased network performance through minimal communication overheads, no processing overhead at core routers and easily deployable information.

The selection of the candidate paths is a key factor in Localised QoS routing and has a considerable impact on how the Localised routing algorithm performs. Various methods for the candidate path selection process and their performance can be found in [56] [57].

2.4.2.1 Localised Proportional Sticky Routing Algorithm

The Localised Proportional Sticky Routing algorithm (PSR) [55] was the first Localised 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 $R = R^{min} \cup R^{alt}$.

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 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 $b_r > b^*$, $\gamma_r=\max(\gamma_r-1, 1)$, where ψ is a configurable parameter to limit the 'knock-on' effect [116] under system overloads. Note that $\gamma_r \geq 1$ ensures that some flows are routed along alternative paths to measure their quality.

2.4.2.2 Localised Credit Based QoS Routing

The Localised Credit Based Routing algorithm (CBR) [58] uses simple routing rules to route flows across the network. The CBR scheme performs routing using a crediting scheme for each candidate path in a candidate path set that rewards a path upon flow acceptance and penalizes it upon flow rejection.

The path selection relies on the path's credits: the larger a path's credits, the larger chances for selection to send the flow. The CBR algorithm keeps updating each path's credits upon flow acceptance and rejection and it does not compute a flow proportion as in the PSR algorithm. It periodically keeps monitoring the flow blocking probabilities for each path and conveys the statistics data to the credit scheme to use it in each path for path selection.

A set of candidate paths R between each source destination pair is required in the CBR algorithm. Like PSR, CBR predetermines a minimum hop path set $R^{\min}$ and an alternative path set R^{alt}, where $R = R^{\min} \cup R^{alt}$. Upon flow arrival the CBR algorithm starts to select the largest credit path P.credits in each set of paths, minimum hop path set $R^{\min}$ and alternative path set R^{alt}. 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 credit P^{alt}; otherwise the flow is routed along an alternative path, as described in the following formula (2.3).

$P^{\min}.credits >= \Phi \times P^{alt}.credits$, where $\Phi \leq 1$ .... (2.3)

Φ, is a system parameter that controls the usage of alternative paths. CBR uses blocking probabilities in the crediting schemes to improve the algorithm's performance, as a path with low blocking probability will gain more credits. The path credits are incremented or decremented upon flow acceptance or rejection using statistics of the path blocking probability.

However, the CBR uses a MAX_CREDITS system parameter to determine the maximum attainable credits for each path by computing the blocking probability using formula (2).

$0 \leq$ credits $\leq$ MAX_CREDITS .... (2.4)

The 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 0s by M to estimate each path's blocking probability for a period of M connection requests.

2.5 Time Complexity

There are some problems associated with QoS routing algorithms in large-scale networks, such as complexity, scalability and optimality [11]. It should be possible to scale QoS routing algorithms to large networks, however this introduces more complexity and more overheads in order to find a feasible path. Time complexity relies on the number of computational steps that an algorithm takes to find a solution for a given QoS routing algorithm problem. So, the time complexity of a QoS routing algorithm is related to the number of operations needed to satisfy QoS and its composition rules. Table 2.3 lists the time complexities for global and local routing algorithms. In QoS source routing algorithms, time complexity is a major performance criterion, since all computational steps to find a feasible path are carried out by the source [45].

Table 2.3 Time complexity of QoS routing algorithms

Routing Algorithm	**State-Information**	**Time Complexity**
WSP	Global	O ($N \log N+L$)
SWP	Global	O ($N \log N+L$)

SDP	Global	O ($N\ log\ N+L$)
Dijkstra	Global	O ($\lvert N \rvert^2$)
PSR	Local	O (R)
CBR	Local	O (R)
N = number of nodes, L = number of links , R = number of candidate paths		

2.6 Summary

The benefit of QoS-routing in packet switched networks such as the Internet is undeniable, it brings about additional complexity and the diverse QoS requirements of many emerging applications are calling for a new paradigm beyond the best-effort approach traditionally used by the Internet. This thesis provides an overview of the QoS paradigm and its main components.

Achieving scalability is a major concern for QoS routing, and this problem is persistent. As the Internet grows, so does the global state, state information inaccuracy and the routing costs. The most important strategies for QoS routing are included, along with their advantages and drawbacks, such as source and distributed routing, but they are short-term solutions. Seemingly, source routing is more favoured than distributed routing in many solutions for QoS-routing. Perhaps the main reasons for this are because source routing supports many legacy protocols (i.e. OSPF) with minor modifications and best achieves the last two objectives of QoS-routing in Section 2.3.

The most popular approaches to QoS routing presented with their related algorithms. Each algorithm tries to tackle the problem of QoS routing in a different way according to its goal. The algorithm proposed by Guerin et al. in their extension (QOSPF) uses this technique to pre-compute minimum hop paths for all destinations and a set of quantised bandwidth values.

Most QoS routing algorithms try to find the best path for each destination and send flows to their destinations over the best path by using a global view of network traffic. A different approach was proposed by Nelakuditi to provide a higher quality of service. Localised QoS routing is a novel approach that attempts to circumvent problems associated with maintaining global state information.
Since routers generally do not have a complete overview of network traffic, the only feasible way to split traffic over multiple routes is to use flow statistics collected locally.

Chapter 3

Simulation Model and Performance Measures

3.1 Introduction

In this chapter, a complete simulation model of QoS routing is developed in order to assess the performance of the proposed algorithms. The simulation model will also enable us to compare the proposed algorithms under different network configurations. Based on this model, extensive simulation experiments will be carried out using various types of network topologies, traffic patterns and under a wide range of traffic loads. The simulation was developed for emulating realistic packet switched networks, such as the Internet, under different scenarios. The simulation model mainly consists of three parts:

- The simulator package: this contains implementations of Localised QoS routing algorithms which can be used to evaluate their performance in a realistic manner.

- Simulation performance metrics: This is a set of performance evaluation metrics that can be collected and reported by the simulator package.
- A set of scenarios and simulation configurations that can be used in other studies to evaluate the performance of new Localised QoS algorithms.

The following sections describe this simulation model in detail. However, since routing is a network layer (i.e. higher layer) entity and due to the dynamic nature of the current realistic network topologies, the input to the routing algorithm will simply be the state information and the topological structure of the underlying network. This chapter discusses some adopted simulation parameters and conditions for simulating the proposed algorithms in the next chapters. It focuses primarily on aspects of network models, simulations and performance measures.

3.2 Simulation Environment

The performance of the proposed algorithms are evaluated using the open-source simulation environment OMNeT++ [60] [61]. OMNeT++ (Objective Modular Network Testbed in C++) is a discrete event simulator that is highly modular, well structured and scalable. It provides a basic infrastructure wherein modules exchange messages and provides a set of functions and tools for simulating the elements of a communication network, such as nodes, links and packets. An

OMNeT++ simulation is built on C++ foundations using the simulation class library. Modules are built out of hierarchically nested modules in which the number of hierarchy levels is not limited. Modules are programmed in C++ and use messages as the means of communication with each other. A node maintains an arbitrary number of gates that are used to send messages through links to other nodes. A network topology that contains gates, links and modules, is defined by the Network Description language (NED language).

3.2.1 Simulator Design

The modelling of large-scale networks is an issue that needs to be addressed in designing a network simulator environment. Moreover, the traffic arrivals of very large numbers of flows (over two million) are simulated in order to give accurate statistical results and to emulate realistic modern networks, which have very large capacities and can accommodate very large numbers of flows.

With these goals in mind, It is found that the ns-2 simulator package focused on low level modelling, such as packet-level details and would require considerable customisation and the addition of missing components. Moreover, due to the outstanding number of simulation events that grow linearly with the numbers of concurrent flows, this package suffers from scalability problems and this could lead to performance bottlenecks when managing a sorted event list of millions of events. Hence, a general-purpose network simulator is chosen to provide the basic functionalities for modelling the network elements and would enable us to develop the key components of our simulator.

For this reason, our simulator is developed on top of OMNET++ [61] [62] which is a discrete event-driven simulator that provides a rich set of functions and tools for simulating the elements of a communication network, such as nodes, links and packets.

3.2.2 Simulator Structure

The functional diagram of the Localised simulator is illustrated in Figure 3.1. The functional diagram shows the key model components of the Localised simulation procedure. In particular, the diagram gives an overview of the simulation process and how it performs a specific task, as illustrated below.

Simulation Initialisation and Network Topology Setup

The module with its variables performs initialising steps in the entire simulation process. This includes reading and setting up the network topology, link capacities and simulation parameters. In addition, the set of candidate paths for each source and destination pair is constructed based on topology information parameters, this set is used by Localised algorithms only.

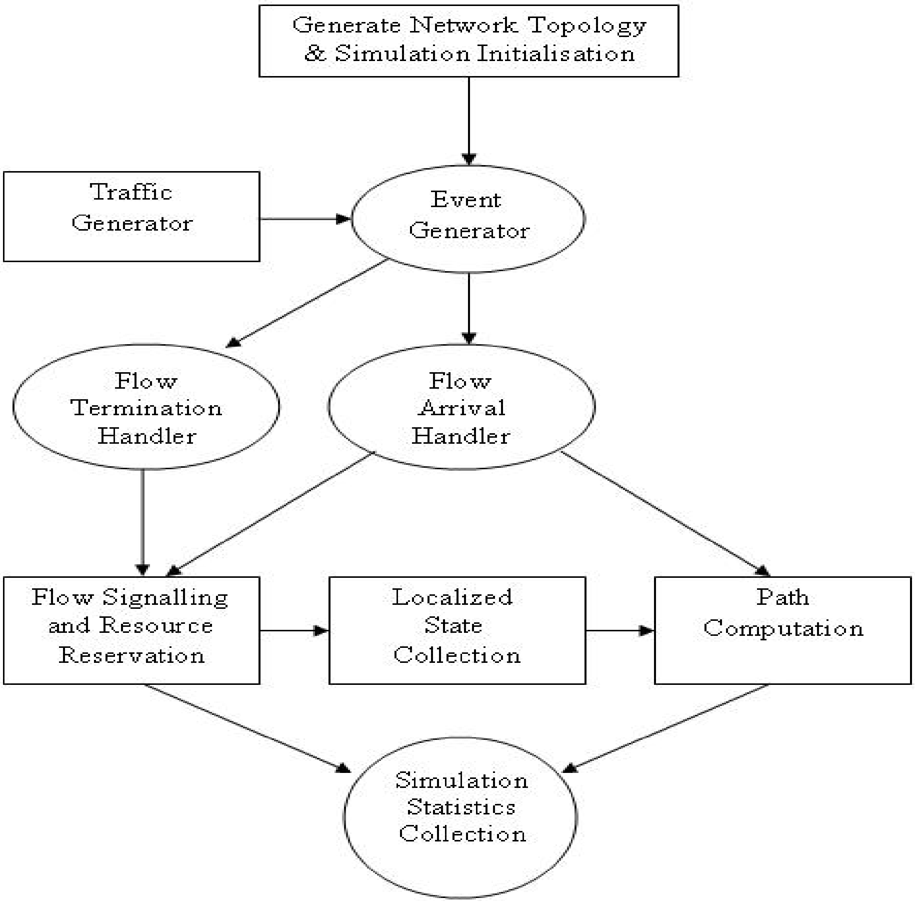


Figure 3.1 Functional components of Localised simulator

Traffic Generator Module

The traffic generator module determines the characteristics of the traffic in the network, based on traffic parameters. Upon flow arriving to a source node the traffic generator provides the flow with a random destination node based on the flow duration. The traffic generator module supports the following characteristics:

- Flow arrival process: the traffic can be modelled as a Poisson stream or bursty stream with variable shaper.
- Bandwidth requirements: either uniformly distributed or fixed.
- Source-destination selection: chosen randomly.

Event Generation

The event generation can perform the following events:

- **Flow Arrival**: The arrival of a flow will trigger the invocation of this handler which will pass each flow along with its bandwidth requirements to the path computation module to determine the best seemingly feasible paths for that connection; once the feasible path is determined the handler will invoke the flow signalling resource reservation to signal the flow.
- **Flow Termination:** the flow termination handler releases the reserved resources to terminate the flow using the flow and signalling resource reservation module.

Path Computation Module

This module implements various routing policies for Localised QoS routing algorithms. The algorithms use the information collected by the Localised state collection module to route traffic through the most appropriate candidate path.

Flow Signalling and Resource Reservation Module

This module simulates the basic flow signalling and resource reservation mechanism policy. When the path computation module returns a path that is feasible and which has satisfied the QoS requirements, the signalling mechanism initiates hop-by-hop signalling by the source node to reserve network resources for each connection arrival.

- *Bandwidth (Requested QoS)*

The signalling mechanism is simulated where the source node initiates hop-by-hop signalling to reserve the requested bandwidth b on each link l in the network, l has the available bandwidth BW (l) which can be allocated to new flows.

As the signalling message traverses through the selected path p, each node carries out an admission test to check over the outgoing link to make sure that the flow actually has sufficient bandwidth.

If the available bandwidth over the outgoing link has sufficient resources, where the available bandwidth is equal to or greater than the requested bandwidth, the node reserves the bandwidth b for the new flow, so that (i.e. BW (l) = BW (l) – b) before passing the set-up message to the next link in the path. This module accepts the flow and the flow is admitted into the network if all links along the selected path p have enough bandwidth to support the flow, otherwise, a failure message is transmitted back to the source node releasing the previously reserved bandwidth (i.e. BW (l) = BW (l) + b) and the flow is rejected.

Localised State Collection

This module is responsible for collecting statistics about network state required by these algorithms and supports Localised QoS routing algorithms. Note that this module interacts with the flow signalling and resource reservation module in order to infer information about the network state and obtain information regarding a flow's acceptance and rejection.

Simulation Statistics Collection

This module monitors large numbers of information statistics during the simulation runs. Locally collected flow statistics information is used by the source node to generate flow arrival rates, flow departure rates and routes traffic. This also includes performance metric statistics information, such as flow blocking probability, bandwidth blocking probability and average carried traffic. The statistics collection module is invoked by different simulator modules in order to collect different aspects of the performance metrics.

3.2.3 Performance Metrics

3.2.3.1 Blocking probability

The goal of QoS routing is to find a route from a source to a destination that satisfies the QoS requirements. In the flow-based model with bandwidth guarantees, a flow is admitted if it satisfies the required QoS along the path through which it is routed. If the QoS is not satisfied the path cannot be used and so the flow is rejected. Hence, only need to perform routing at flow level simulation to study the performance of different QoS routing schemes. The objective of QoS routing is then to minimise the blocking probability experienced by a flow arriving in the network.

Valuable network resources have been used in path computation and, therefore, this is an undesirable overhead associated with a QoS routing algorithm. Ideally, it is needed to have a lower blocking probability with a minimal overhead. This overhead is measured by the ratio of the number of paths rejected to the total number of flows during long time intervals. This gives an estimation of the flow blocking probability which is, therefore, also a measure of the efficiency of the QoS routing scheme used. Essentially, the objective of QoS routing is to minimize the overhead while minimising the flow blocking probability. The flow blocking probability is used as the main performance metric to measure the performance of the proposed algorithms. The flow blocking probability is defined as:

$$\text{Flow blocking probability} = \frac{\text{Number of rejected requests}}{\text{Number of requests arriving}} \quad \ldots\ldots\ldots\ldots \quad (3.1)$$

Flows will be rejected when one of the links along the path from source to destination does not satisfy the requested bandwidth. The notion of bandwidth blocking probability also is used to solve discrimination against flows with large bandwidth requirements. The bandwidth blocking probability is defined as:

$$\text{Bandwidth blocking probability} = \frac{\sum_{i \in B} bandwidth(i)}{\sum_{i \in c} bandwidth(i)} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3.2)$$

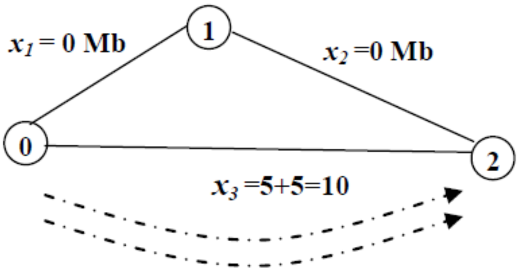
Here B is the set of blocked flows and C is the set of total flows. Bandwidth (i) is the requested bandwidth for path i.

3.2.3.2 Load balancing

Load balancing is considered to be one of the most important factors in evaluating the performance of QoS routing algorithms [96]. The importance of the blocking probability metric has been overshadowed, since most previous studies in QoS have not included a quantitative metric for this criterion.

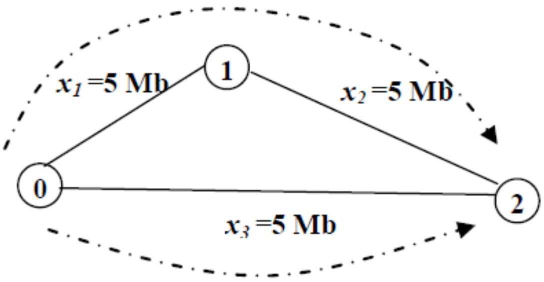
The objective of load balancing is to use the network resources more efficiently in order to reduce the network traffic congestion. Originally, it was important in designing QoS routing algorithms to measure the fairness and distribute the load among links in network topologies. One of the important metrics of evaluation of a QoS routing scheme is, therefore, load balancing where the fairness of any resource allocation is computed in the network to measure the efficiency of routing algorithms.

The well-known Jain's fairness index [63] is used as a measure of load balancing of the algorithms. Let N be the number of links in the network; Jain's index is defined for a set of values $\{x_1, x_2,.., x_N\}$ as:



Bandwidth usage= 0+0+10= 10 Mb

(1) Example 1



Bandwidth usage= 5+5+5= 15 Mb

(2) Example 2

Figure 3.2 Jain's index illustration

$$Jain's\,Index = f(x_1, x_2, \ldots, x_N) = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N\sum_{i=1}^{N} {x_i}^2} \qquad \ldots\ldots\ldots\ldots\ldots\ldots \qquad (3.4)$$

Jain's index is bounded between 1/N and 1: when the value is 1 the algorithm is 100% fair, whereas a value of 1/N is the least fair.

To explain the Jain index usage in the context of QoS routing algorithms the two situations are considered as illustrated in the example 1 of Figure 3.2, where it is possible to setup two connections from node 0 to node 2, each with 5 M bits of bandwidth. In the first case, the two flows were routed along the direct link, while in the second case, they were split to balance the load among the links. The Jain index in the first case is 0.33 which means less fairness compared to the second case that has a Jain index fairness of 1, which means complete fairness. Note that balancing the load may cause inefficient resource usage optimisation as a result of using longer paths to avoid congested links, as shown in the example 2 of Figure 3.2, so the two optimality constraints conflict with each other and a trade-off situation must be sought [35] [52] .

3.2.4 Network Topologies

A key challenge in studying routing algorithms in wide area networks lies in how to represent the underlying topology given the constantly changing and decentralised nature of the Internet [65] [66]. The characteristics of these topologies have not been well understood and this makes it hard to define a "typical" topology [67]. Adding to the challenge are observations that conclusions about performance of routing algorithms may in fact vary dramatically with the underlying network topology.

For example, random graphs can result in unrealistically long paths between certain pairs of nodes; “well-known” familiar topologies may show effects that are unique to particular topologies [2]. Consequently, simulation experiments in this study consider a wide range of network topologies which represent different topologies with different characteristics.

ISP Topology

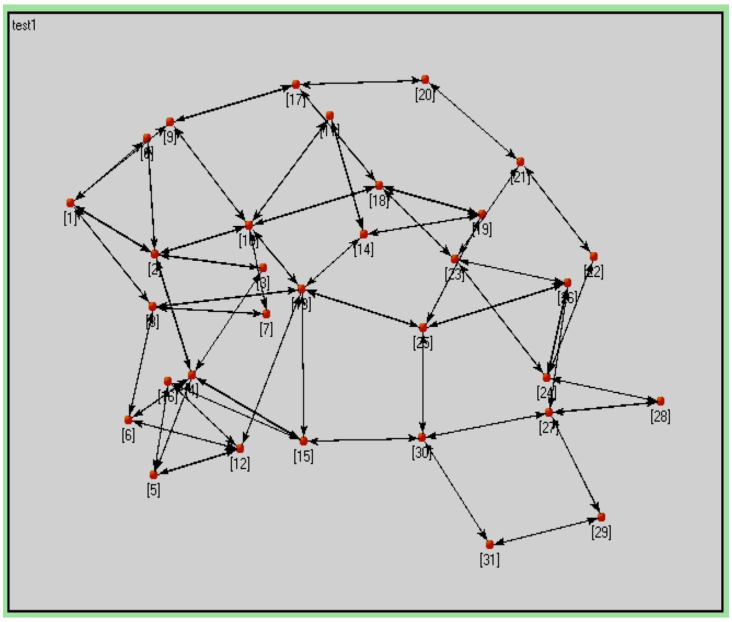


Figure 3.3 ISP Topology

The familiar ISP topology is studied which has appeared extensively in simulation of routing algorithm studies [68] [69] [72]. The ISP network topology, shown in Figure 3.3, can also be considered as a single autonomous system domain for Internet Service Provider’s networks [93] and it is typical of the nationwide network in the USA. Regular topologies are not considered such as torus and cube because they are infrequently used in realistic networks [73]. Although the structure of regular topologies makes them analytically tractable, the properties of such configurations are quite different from the properties measured on actual networks [65].

Scale-Free Topology

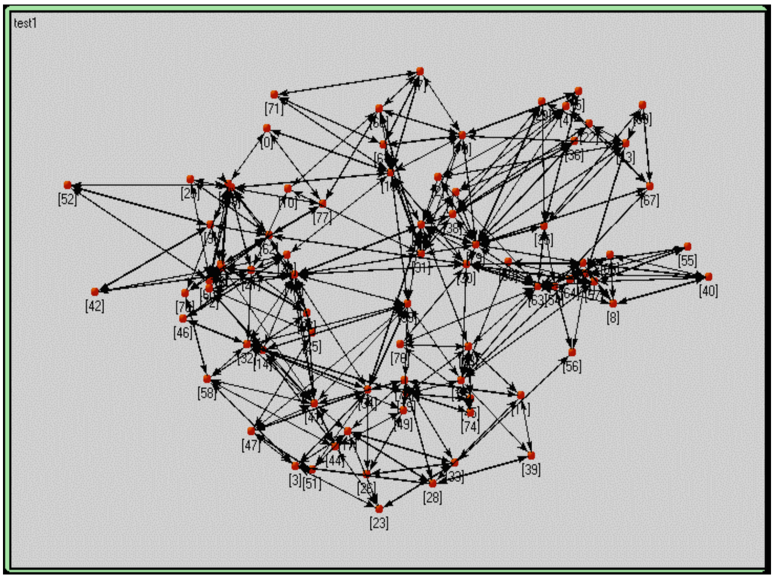
In scale-free [105] graphs, node degree distribution is characterised by power-law distribution [106]. Most nodes have small link connections while the remaining few have a high degree of link connections acting as highly connected hubs. Average path length in a scale-free graph is small, and this is called the small world property [103], [105]. Scale-free graphs, as opposed to random graphs, are in fact a true resemblance of the Internet network topology. Scale-free topology is generated on top using the BRITE generator [70] using the Barabási and Albert model [104].

Name	Nodes	Links	Avg. degree	Diam.	Avg. path length
RANDOM80	80	480	6	6	3.008
ISP	32	108	3.375	6	3.177
RANDOM 32	32	122	3.812	4	2.415
SCALE-FREE	32	122	3.812	2.4274	10.179

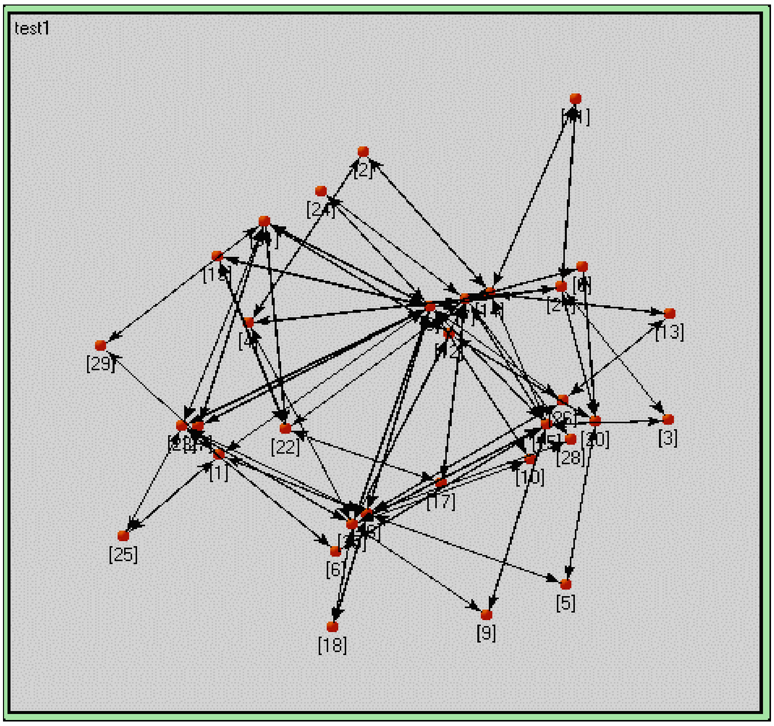
Table 3.1 Characteristics of selected topologies

Random Topologies

Rather than considering regular topologies, which may hide important effects of heterogeneity and non-uniformity, random topologies are proposed. Random topologies are generated on top using the BRITE generator [70] using Waxman's model [71]. Table 3.1 lists the most important characteristics of the selected topologies used in the experiments, where Figure 3.4 shows the ISP and random topologies.



(1) Random 80 Topology



(2) Random 32 Topology

Figure 3.4 Selected random topologies used in simulations

3.2.5 Simulator Validation

Validation guarantees that the simulation behaves as expected to ensure that there are no significant differences between the simulator model results and ones known to be correct [64]. Simulator model validation is the method used to prove that the results obtained from simulation experiments are correct. For this thesis, some validation testing has been conducted. Simulations are built on top of the OMNeT++ simulator to verify the correctness of the proposed Localised QoS routing algorithms. The core simulation engine is assumed and the functionality provided by OMNeT++ has been properly validated, since it has been used in the research community for a long time and results of many researchers have been achieved using this simulator. More information can be found in [60].

Regarding validation of the proposed Localised QoS routing algorithms, the results obtained have been compared to those obtained by the credit based routing algorithm (CBR) [58]. For the CBR algorithm, using the same simulation parameters and configurations described by the designer of the CBR algorithm [58], the simulations are repeated using the simulator and found that the results were very close to the results reported.

All results were collected for all simulation experiments from at least 2,000,000 connection request arrivals and the results were collected after 200,000 connection requests to allow a steady state to be reached. Each simulation experiment was repeated 20 times with different seeds; the 95% confidence intervals were computed and found to be extremely tight, such that in most figures only mean values of the results were presented.

3.3 Summary

In this chapter, the simulation model is presented which has been built on top of OMNeT++, through which to evaluate the performance of the proposed algorithms in the following chapters. In this chapter the simulator design was described, along with details provided as to how it was validated. Also, the network topologies, parameter settings and the performance metrics, such as blocking probabilities and load balancing, used in the performance evaluation were described.

The attention next concentrated on local state information. In the following chapters a novel Localised routing algorithm is introduced to tackle the two major overheads of QoS routing in a very simple yet effective manner.

Chapter 4

A new Localised Quality Based Routing Algorithm

(Bandwidth as a new Path Selection Routing Method)

4.1 Motivations

Routing in packet switched networks, such as the Internet, typically supports only best-effort traffic which was not originally planned for optimizing the performance of the network. The routing protocols deployed, such as OSPF [74] [75] [76], use the shortest path only routing mechanism. The OSPF protocol in the Internet, its extension QOSPF and the PNNI protocol of ATM networks [77] are practical examples of global algorithms which require the maintained knowledge of global state information to compute the QoS path.

Although these protocols are well suited for traditional data applications, such as e-mail, ftp and telnet, they are inadequate for many emerging multimedia applications, such as video on demand and teleconferencing, which require bandwidth guarantees. Furthermore, the global state information for the complete network state in each node suffers from the scalability of QoS routing algorithms in large networks.

Moreover, with the current exponential growth of Internet traffic, the shortest only routing mechanism leads to unbalanced traffic distribution, since links become increasingly congested, while links not on the shortest path are underloaded. The Localised approach has recently been proposed as a viable alternative approach in which routing decisions are based on local information collected by the source node. A fundamental difference between the two approaches lies in the amount of network information needed to perform routing. Global network state information is large, especially for large networks, and it changes rapidly with every flow arrival and departure.

The Localised QoS routing schemes avoid the overhead messages that are exchanged in order to keep up-to-date global state information and, therefore, avoid the routing scalability problem which occurs in large-scale networks. In Localised routing, only a small set of candidate paths needs to be maintained at each node, along with their locally collected statistics. In contrast to the global state which changes rapidly with network load, paths in the candidate set are selected based on relatively static information, such as topological information.

The success of Localised routing in telephone networks [79] and the limited number of promising researches in the area of Localised QoS routing (in the context of packet switched networks) has motivated the present author to search for other viable novel Localised methods.

The Localised proportional sticky routing (PSR) algorithm [54] [55], was the first Localised QoS scheme. Since it has been shown that the CBR algorithm outperforms the PSR algorithm [2] [78] in almost all situations. This study subsequently focusing the attention on using the CBR algorithm for benchmarking; the CBR scheme [78] is the most relevant work to this study

Although, CBR exhibits a good performance against the PSR algorithm and it was performing at a more enhanced level than global WSP algorithm [47] the criteria used for path selection in the CBR algorithm relies on a crediting criteria, it does not directly reflect the quality of a path, which is indirectly reflected by the addition or subtraction of credits criterion. It is conjectured the quality of a path should be measured directly based on the path selection routing and also required to use bandwidth in CBR as the QoS metric for path selection. Moreover, there does not appear to be any analytical justification to compute blocking probability as an increment or decrement factor for the credits, but the justification appears entirely intuitive. The CBR algorithm is not a gradient variable using manipulation by blocking probability.

In this chapter a Localised acceptance rate based routing algorithm (ABR) is introduced which directly uses an average statistics to make routing decision. This is then a novel quality based routing algorithm (QBR) is developed which is an effective Localised QoS routing algorithm, in order to use a bandwidth (the same QoS metric used) as a path selection routing metric. The Localised credit based routing (CBR) proposed in [78] is studied further and compared its performance with the proposed schemes, in terms of flow and bandwidth blocking probabilities under various ranges of traffic loads and different network topologies.

4.2 Localised Acceptance Rate Based Routing (Average Path Statistics as a Path Selection Algorithm)

Unlike the CBR algorithm, which critically relies on a path’s credit, the Localised acceptance rate based routing (ABR) uses simple routing rules across the network, using the acceptance rate of the path without a crediting scheme, in order to make routing decisions. It rewards a path upon flow acceptance and penalizes it upon flow rejection.

The ABR is a source routing algorithm where the source node makes the routing decision. When a new flow arrives, the path is computed by the source node which is likely to satisfy the requested QoS. This path selection is performed by the source node based on its own local view which is updated from previous routing attempts. Once a path is selected for the flow, the source node sends a setup message to travel along the selected path with each connection request. Each intermediate node performs an admission test for outgoing link residual bandwidth to see whether the link has the ability to support this flow. If the link can accommodate the flow, the requested bandwidth is reserved for that flow and the setup message is forwarded to the next hop along the path. The network therefore admits the flow if all links along the path can satisfy the requested bandwidth and the setup request is accepted. The resources reserved for a flow remain with it for the entire flow duration. However, if any link along the path does not support the requested bandwidth, the setup message is categorised as a setup failure which is rejected and the flow is blocked. Messages statistics are sent to the source node to calculate the acceptance rate from path request to path request.

The pseudo code for the ABR algorithm is given in Figure 4.1. Like CBR, ABR requires every node to maintain a predetermined *set of candidate paths R* for each possible destination. ABR distinguishes between two types of paths, the set of minimum-hop paths $R^{\min}$ and the set of alternative paths R^{alt}, where $R = R^{\min} \cup R^{alt}$. Every path P associated with a variable P.Rate stores the accumulated values gained so far. Upon flow arrival, ABR selects two paths, $P^{\min}$ (line 3) and P^{alt} (line 4) which are the paths with maximum acceptance rate in $R^{\min}$ and R^{alt}, respectively.

The flow routed along $P^{\min}$ if: $P^{\min}$ Average $\geq \Phi \times P^{alt}$ **.** Average (lines 5-6), where $\Phi \leq 1$, otherwise, P^{alt} is chosen (line 8). Φ is a system parameter that controls the usage of alternative paths. Note that if there is more than one path with the same acceptance rate the first one is chosen.

When the flow is accepted along the selected path (line 10), 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.

```
Initialize

  Set $P.Rate=1, \forall\ P \in R$

ABR
  if $P.Rate=0\ \forall\ P \in R$
    set $P.Rate=1,\ \forall\ P \in R$
  $P^{min}=\max\{P.Rate:\ P \in R^{min}\}$.
  $P^{alt}=\max\{P.Rate\ \ P \in R^{alt}\}$.
  if($P^{min}.Rate>=\Phi \times P^{alt}.Rate$).
    set $P=P^{min}$
  else
    set $P=P^{alt}$
  route flow along path P.
  if flow accepted
  UpdatePath'sAcceptanceRate(1)
  Compute P.Rate.
  else
  UpdatePath'sAcceptanceRate(0)
  Compute P.Rate.
```

Figure 4.1 ABR pseudo code

Since the ABR 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. 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 4.1 is the ABR 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 Localised schemes, which indirectly reflect the quality of the path by the addition or subtraction of the credits criterion, the ABR computes the acceptance rate directly.

4.3 Localised Quality Based Routing (Bandwidth as a Path Selection Algorithm)

Unlike the previous Localised schemes, whereby the only available information at the source node entailed statistics regarding blocked flows, in order to make a routing decision the Localised quality based routing algorithm (QBR) is proposed to modify the ABR algorithm from average path's acceptance to average path quality using the same QoS metric (bandwidth) for routing decisions.

QBR aims to select a candidate path where the bottleneck bandwidth is above a required value, since a QBR is a bottleneck bandwidth algorithm. The state of each candidate path is calculated by the state of the bottleneck and then the resulting average value is used to measure the quality of the path. Upon flow arrival the path with the largest average path quality is used to route the incoming flow. QBR keeps monitoring the bottleneck links for each candidate path set in the network and continuously updates each path from path request to path request. By using updates bottleneck link values actually reflect the quality of the path.

The QBR is a source routing algorithm where the source node makes the routing decision. When a new flow arrives, the path is computed by the source node which is likely to satisfy the requested QoS. This path selection is performed by the source node based on its own local view that is updated from previous routing attempts. Once a path is selected for the flow, the source node sends a setup message to travel along the selected path with each connection request. Each intermediate node performs an admission test for outgoing link residual bandwidth to see whether the link is able to support this flow. If the link can accommodate the flow, the requested bandwidth is reserved for that flow and the setup message is forwarded to the next hop along the path. The network therefore admits the flow if all links along the path can satisfy the requested bandwidth and the setup request is accepted. The resources reserved for a flow remain with it for the entire flow duration. However, if any link along the path does not support the requested bandwidth, the setup message is categorized as a setup failure which means it is rejected and the flow is blocked. Messages statistics are sent to the source node to calculate the average path quality from path request to path request.

The pseudo code for the QBR algorithm is given in Figure 4.2 as follows: Localised quality based QoS routing requires a predetermined set of candidate paths R. The main characteristic that is associated with every path P in the candidate path set is the average path quality computed by each candidate path’s bottleneck bandwidth. P.Quality is used to store the average values and update its value with each path request. Upon flow arrival, QBR selects the path with the largest average path quality (line 3) and routes the flow along the selected path.

```
Initialize
    Set P.Quality=1, $\forall$ $P \in R$
    QBR
    1.  if P.Quality=0 $\forall$ $P \in R$
    2.    set P.Quality=1, $\forall$ $P \in R$
    3.  $P^{min}$ =max{P.Quality: $P \in R^{min}$ }.
    4.  $P^{alt}$ =max{P.Quality  $P \in R^{alt}$ }.
    5.  if($P^{min}$.Quality>= $\Phi \times P^{alt}$.Quality).
    6.    set P=$P^{min}$
    7.  else
    8.    set P= $P^{alt}$
    9.  route flow along path P.
    10. if flow accepted{
    11. if(ResidualBandwidth>=RequestedBandwidth)
    12. UpdatePathQuality(1)
    13. Compute P.Quality.
    14. else
    15. UpdatePathQuality(NormalisedResidualBandwidth)
    16. Compute P.Quality.}
    17. else if flow rejected
    18. UpdatePathQuality(NormalisedResidualBW)
    19. Compute P.Quality.
```

Figure 4.2 QBR pseudo code

If the flow is accepted along the accepted path, the residual bandwidth is calculated along it. As the setup message travels to the destination it performs a comparison over the links along the path to get the minimum residual bandwidth for that path (line 11). The QBR algorithm focus on bottleneck links along the path, the source node continues to use the same route unless another path is much better in terms of path quality, if the flow is accepted and the value of a bottleneck link along the path is above or equal to the requested bandwidth the update value is stored at the source node by one (line 11). If the flow is accepted and the value of a bottleneck link along the path is less than the requested bandwidth the update value is stored at the source node by a normalised value less than one (line 15). If the flow is rejected the update value is stored at the source node by a normalised value (residual bandwidth – requested bandwidth) which is less than zero (line 18). The QBR algorithm continuously updates values within intervals of [-1, 1]. The path's update values are therefore added to previous values of the path and stored in the source node.

As a new connection arrives with the source node, the stored values are divided by the number of connections sent in order to estimate the actual average path quality. It should be noted that choosing the minimum residual bandwidth among the links on the selected path reflects the actual bandwidth that the path can support. In contrast, if the flow is rejected its residual bandwidth is subtracted from the overall path residual bandwidth and the new average is calculated as previously. When the path's bottleneck links decrease the probability of being chosen also decreases for new connections. Increasing or decreasing the path bandwidth actually reflects the quality of the path which can be measured accurately.

QBR monitors the bandwidth of a path and the source node stores the accepted or rejected flow of each path respectively. It calculates the average path quality using a simple moving average (sliding window) over a predetermined period. So, for a period of M, the average path quality will be calculated over the most recent M connection requests.

For example, let $S=\{1, 0.6, -1, -0.1, 1\}$ represent the values collected over data from the last predefined period M =5 flows for the path P, the average path quality would be 1.5/5. The oldest element is in the leftmost position whereas the newest one is in the rightmost position. Now, if another flow is accepted, then the oldest element will be deleted from S, and will be replaced by the data from the latest flow.

4.4 Performance Evaluation

To evaluate the performance of the proposed Localised algorithms, the Acceptance rate Based Routing (ABR) and the quality based routing (QBR), they are compared with the Localised CBR scheme. PSR and global QoS routing algorithms were not considered for comparison since Localised CBR has been shown to outperform both PSR and the global QoS routing (WSP) algorithm in almost all situations [2] [55]. In the following, the simulation setup is described first and then compared with the performance of the proposed schemes using the performance metrics defined earlier.

4.4.1 Simulation Setup

The Localised QoS routing schemes ABR and QBR were implemented based on the discrete-event simulator OMNeT++ [60] [61] and extensive simulations were conducted to test their performance. A variety of QoS routing schemes [23] [45] [64] [69] have been proposed to make routing decisions at the flow level. The predetermined Localised algorithms PSR and CBR perform routing decisions in the simulation at the flow level. By performing simulation at the flow level it is possible to simulate the performance of thousands of flows in an efficient manner; consequently the objectives and procedures between flow level and packet level are different. Hence, the flow level simulation is needed to perform studying the performance of QBR scheme.

Due to the fact that performance of the QoS routing algorithms may in fact vary with underlying network topologies [73], simulation experiments in this study consider different types of network topologies as listed in Table 4.1. A familiar ISP topology is used in the simulation experiments, which has been widely used in different QoS routing algorithm studies [68] [69] [72]. In addition, random topologies are investigated and generated on top using the BRITE generator [70] using Waxman's model [71]. Table 4.1 lists the most important characteristics of the topologies used in the simulation.

Name	Nodes	Links	Avg. degree	Diam.	Avg. path length
RANDOM 80	80	480	6	6	3.008
ISP	32	108	3.375	6	3.177
RANDOM 32	32	122	3.812	4	2.415

Table 4.1 Characteristics of network topologies

4.4.2 Traffic Generation

All links are assumed to be symmetric, bidirectional and have the same capacity C (C=150 Mbps) in each direction. The network topology is assumed to remains fixed throughout each experiment in the simulation; hence the effects of link failures not investigated.

Flows arrive to each source node according to Poisson distribution with rate $\lambda=1$ and destination nodes are selected randomly by uniform distribution. Flow duration is exponentially distributed with a mean value $1/\mu$, while flow bandwidth (QoS requested) is uniformly distributed within the [0.1 – 2MB] interval.

Following [1], the offered network load is $\rho = \lambda N\bar{h}b / \mu LC$, where N is the number of nodes, L is the number of links in the network, b is the average bandwidth required by a flow and $\bar{h}$ is the average path length (averaged across all source-destination pairs.). The parameters used in the simulation for CBR are MAX_CREDITS=5 and Φ=1. Blocking probabilities are calculated based on the most recent 20 flows.

A set of candidate paths is chosen such that for each source-destination pair in a selected network topology, the candidate set consists of paths that have at most one hop more than the minimum number of hops. All results are collected for all simulation experiments from at least 2,000,000 connection request arrivals and the results are collected after 200,000 connection requests to allow a steady state to be reached. Each simulation experiment was repeated 20 times with different seeds; the 95% confidence intervals were computed and found to be extremely tight, such that in most figures only mean values of the results are presented, see Figure 4.3(a).

4.4.3 Performance Metrics

The performance metrics used to measure the performance of the algorithms are flow blocking probability and bandwidth blocking probability. Flows are rejected when one of the links along the path from source to destination does not satisfy the requested bandwidth. The blocking probability is defined as:

$$\text{Flow blocking probability} = \frac{\text{Number of rejected requests}}{\text{Number of requests arriving}} \quad \ldots\ldots\ldots\ldots\ldots \quad (4.1)$$

the notion of bandwidth blocking probability is also used to solve discrimination against flows with large bandwidth requirements. The bandwidth blocking probability is defined as:

$$\text{Bandwidth blocking probability} = \frac{\sum_{i \in B} bandwidth(i)}{\sum_{i \in c} bandwidth(i)} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (4.2)$$

Here, B is the set of blocked paths and C is the set of total requested paths, and bandwidth (i) is the requested bandwidth for path i.

4.4.4 Simulation Results

4.4.4.1 Blocking Probabilities

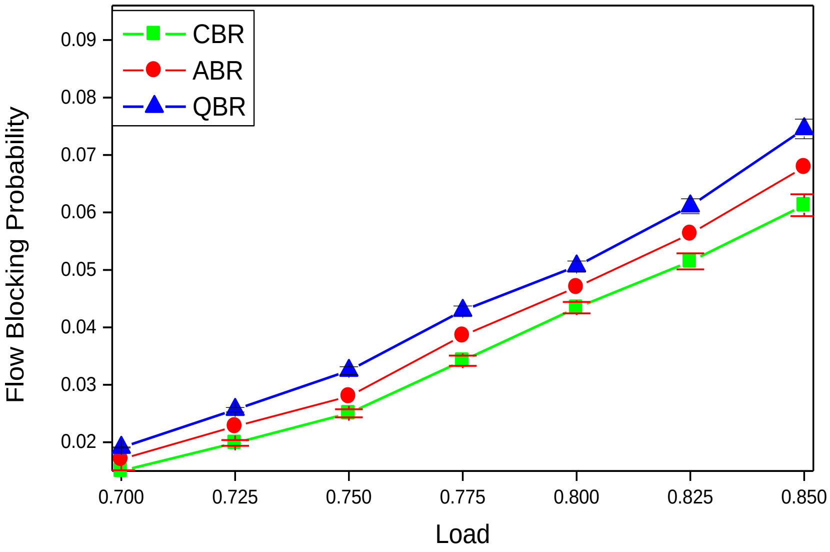
The performance of CBR, ABR and QBR is compared under different settings using the flow blocking probability and bandwidth blocking probability.

4.4.4.2 Various Load Conditions

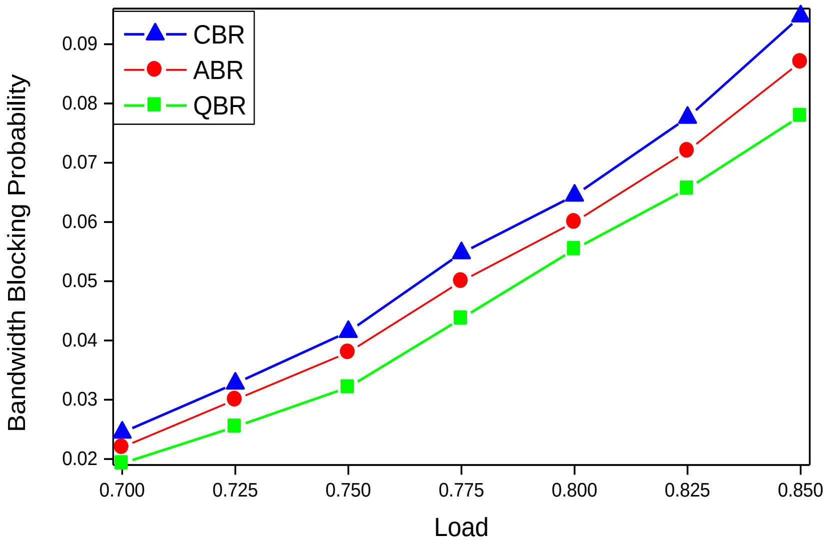
There are several conclusions that can be shown from Figure 4.3. First, there is a marked reduction in the blocking probability by the ABR algorithm when removing the crediting scheme from the CBR performance evaluation. Second, it is evident that there is quite a significant performance gain in using the bandwidth as a path selection metric, instead of only statistics gathered by the source node. Third, in the QBR algorithm, the source node continuously has an advantage in using the same set of candidate paths unless the other path is much better in terms of path quality, which gains the best performance compared to others.

Figure 4.3 compares the performances of CBR, ABR and QBR algorithms in terms of flow blocking probability and bandwidth blocking probability which are plotted under various ranges of offered loads. The flow and bandwidth blocking probabilities are plotted against the offered load using the random topology Random 32. As noted in the Figure 4.3 that under very low loads the performance comparison of the three algorithms is relatively small, which is intuitively expected since the probability of finding sufficient resources in each link is high and flows are almost always accepted. However, performance varies significantly when the load increases, and the bandwidth blocking probability grows more rapidly than the flow blocking probability, implying that flows with large bandwidth requirements are difficult to route, as expected. It is also notable that the relative performance of the three algorithms is the same for flow and bandwidth blocking probabilities, suggesting that either may be used to evaluate the performance of Localised QoS routing algorithms, especially when the bandwidth requirements are slightly small compared to link capacities. Since QBR makes routing decisions based on the bandwidth, which is monitored frequently, its gives the best performance compared with the other algorithms. It even performs better than CBR, particularly under heavy loads.

CBR selects the path using the crediting scheme which is changeable according to the blocking probability, this tends to select relatively more alternative paths. Since credits of the selected path are updated after every path request and routed along that path, any flow rejection will cause its credits to be decreased and an alternative path with more credits to be selected. In the same way QBR selects paths with the largest average path quality using bandwidth, which gives more scope to select paths as long as they satisfy QoS bandwidth.



(a) Flow blocking probability



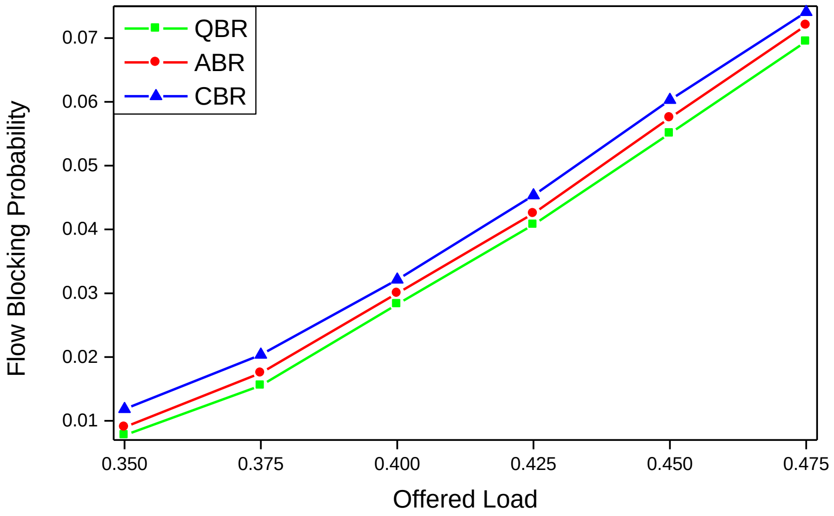
(b) Bandwidth blocking probability

Figure 4.3 Flow and bandwidth blocking probabilities in Random 32

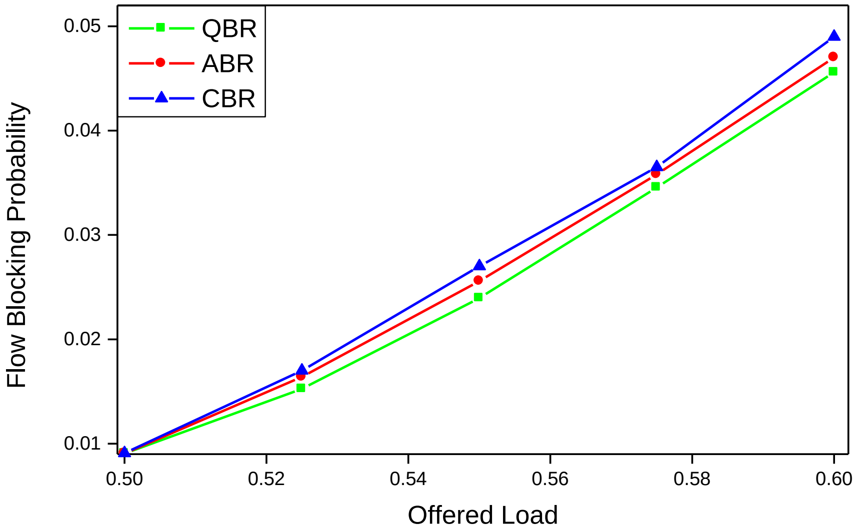
4.4.4.3 Network Topologies

In view of the fact that the performance of any routing algorithm may vary dramatically with the underlying network topology, different types of network topologies were used to evaluate the performance of the algorithms. However, the goal is not to provide a thorough evaluation of the impact of network topology on the performance of routing algorithms, rather, the aim is to illustrate that the algorithms maintain good performance across different topologies. The research direction of impact of topology on the performance of Localised QoS routing algorithms is illustrated in the following chapter. Consequently, simulation experiments consider a range of network topologies with differences in their parameters, such as average path length, number of links between nodes, and average node degree.

Figure 4.4 shows the flow blocking probability for the three algorithms using topologies described earlier in Table 4.1.



(a) Random 80 topology



(b) ISP topology

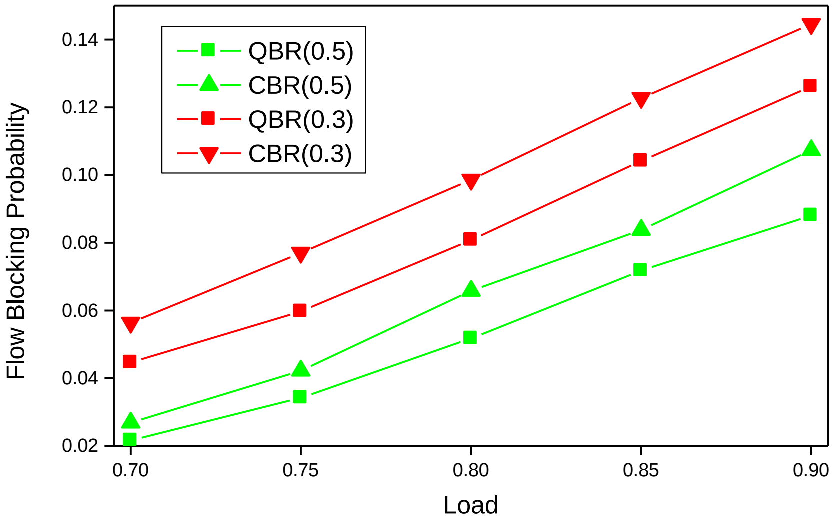
Figure 4.4 Impact of network topologies

Figure 4.4 show that the ABR algorithm and the QBR algorithm give comparable performance against CBR algorithm in all types of network topologies, especially in random topologies.

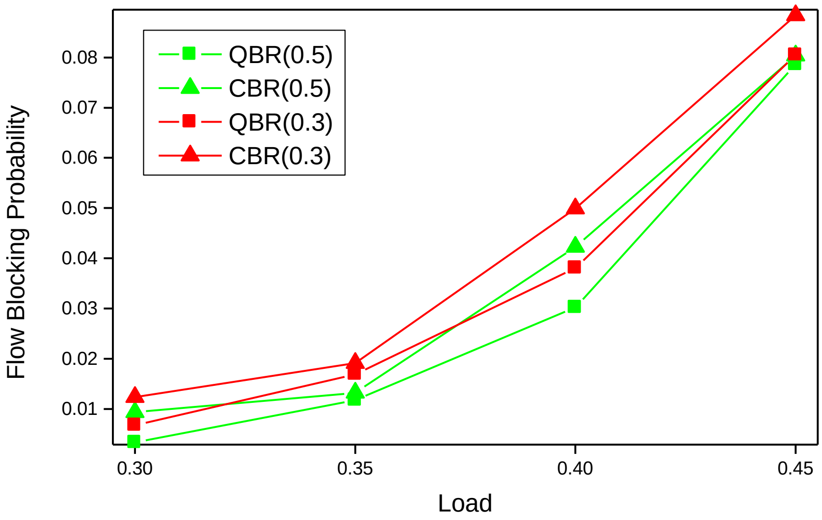
It has been shown that Localised algorithms discriminate against alternative paths which prefer the shortest paths. These algorithms usually outperform algorithms that do not take path length into consideration [1] [64] [73]. The Localised algorithms typically perform well in dense networks (i.e. Random 32 topology) in which there are enough candidate paths between any source and destination. On the other hand, these algorithms are hard to use scarce resources in the sparsely connected ISP topology in which the performance between the proposed Localised algorithms was reduced, as shown in Figure 4.4.

4.4.4.4 Impact of Bursty Traffic

Although, Poisson traffic models have been extensively used to characterize the various dynamics of the Internet flow arrivals, recent studies [94] [95] have showed that the flow arrivals with heavy tails, such as Weibull distributions, give more realistic traffic model representations. Consequently, the effect of bursty traffic is evaluated under the Random 32 topology and Random 80 topology where flow inter-arrival times follow the Weibull distribution.



Random 32 Topology



Random 80 Topology

Figure 4.5 Impact of bursty traffic in random topologies

Following [2] [94], the burstiness of the traffic are modelled with two different values of the shape parameter of the distribution, namely, a=0.3 for the large shape parameter and a=0.5 for the small shape parameter where burstiness is increased with a smaller shape value. Figure 4.5 shows the flow blocking probability plotted against the offered load with different shape values for both random topologies, Random 80 and Random 32.

As shown in Figure 4.5, increased burstiness in the arrival process results in increased blocking probability over the offered loads used. This is in contrast to the CBR algorithm [58], where the burstiness has no significant impact compared with the PSR algorithm, particularly in the case of shape parameter 0.3, where the burstiness is increased. CBR makes routing decisions based on the crediting scheme, which is increased with burstiness. Figure 4.5 show that burstiness has a very significant impact on the performance of the QBR algorithm using different random topologies.

4.4.4.5 Large Bandwidth Requests

To examine the effect of large bandwidth requests, a case of mixed traffic is evaluated using two classes of flows, small-bandwidth flows and large bandwidth, but having the same flow duration. The amount of bandwidth requested by the flows is derived randomly from the range [0.1-2MB], with a mean b_1=1.05 for small-bandwidth flows, and [2-4] with mean b_2=3, for large-bandwidth flows. The flow duration for both bandwidth flows using the Random 80 topology was drawn from an exponential distribution with mean 140 time units. The performance is measured by varying the mix of small and large bandwidth flows, *f* and *(1-f)* respectively, while keeping the offered load fixed. The arrival rates are changed according to the formula:

$$\rho = \frac{\lambda(f \times b_1 + (1-f) \times b_2)Nh}{\mu LC} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (4.3)$$

Figure 4.6 shows the bandwidth blocking probability plotted as a function of the fraction of small-bandwidth flows. As shown in the Figure 4.6 the CBR scheme gives the worst performance regardless of the fraction of small-bandwidth flows. For all schemes, the performance degrades as the fraction of large-bandwidth flows increases. Consequently, as the fraction of small-bandwidth flows increases the blocking probability decreases and again, as shown in Figure 4.6, QBR performs better than CBR with small flows.

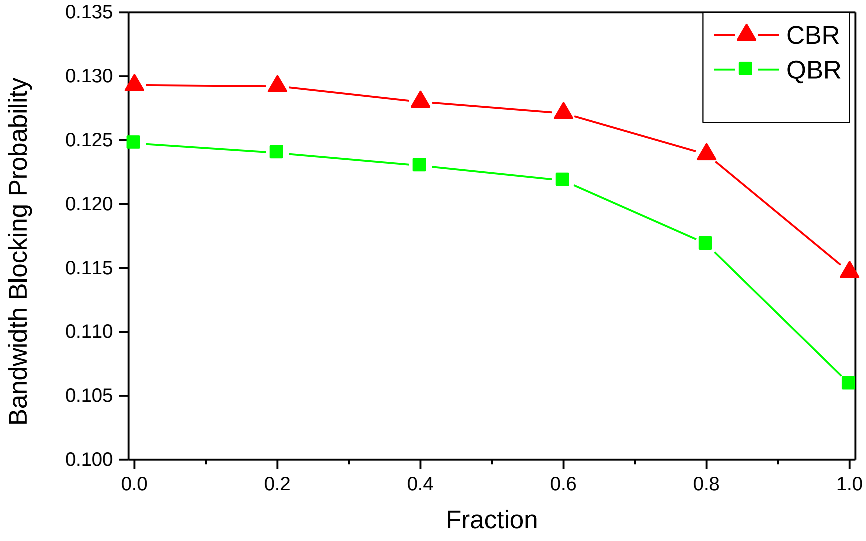


Figure 4.6 Impact of large bandwidth in Random 80.

In contrast to the CBR algorithm, in which the routing is independent of the amount of bandwidth requests, the QBR continuously monitors the bandwidth and the amount of bandwidth flow is stored by the source node before path selection for both small and large-bandwidth requests. However, during the routing process the source node is unable to put the bandwidth requests into classes; therefore it is more difficult to route large-bandwidth flows than to route small-bandwidth flows. It can be noticed that the difference in performance remains fixed between CBR and QBR, and they have good performance with small requested flows or large requested flows, but not with a mixture.

4.4.4.6 QBR Sensitivity to Input Parameters

Since QBR monitors the residual bandwidth in each path among the candidate path set, a predetermined period M (sliding window) is used to record residual bandwidth upon flow acceptance or rejection. Figure 4.7 shows the flow blocking probability plotted against the offered load using different periods of M connection requests. It can be seen that the blocking probability decreases as the value of M increases. M parameter controls the observation period of the path bandwidth, so a longer period is needed to get a good estimation of how good or bad the path is. Figure 4.8 shows the blocking probability plotted against window size M connection requests on Random 32 topology, with the fixed load = 0.8. QBR showed a better performance as the M increases. That is, the more data that is collected about the residual bandwidth, the more accurate the information obtained about the quality of the path. Figure 4.8 shows that the blocking probability sharply decreases as the M increases until it reaches 50. Choosing values for the M between 50 and 200 connection requests, gives a gradual decrease in the blocking probability; whereas values larger than 200 have a negligible effect on the blocking probability. It should be noted that the M in this experiment is appropriate for the Random 32 network topology and may not be the same for other topologies.

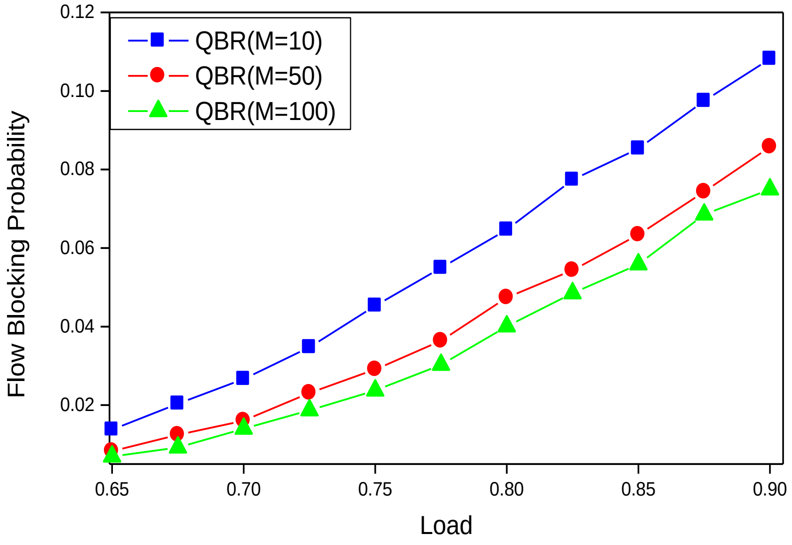


Figure 4.7 Choices of window size in Random 32

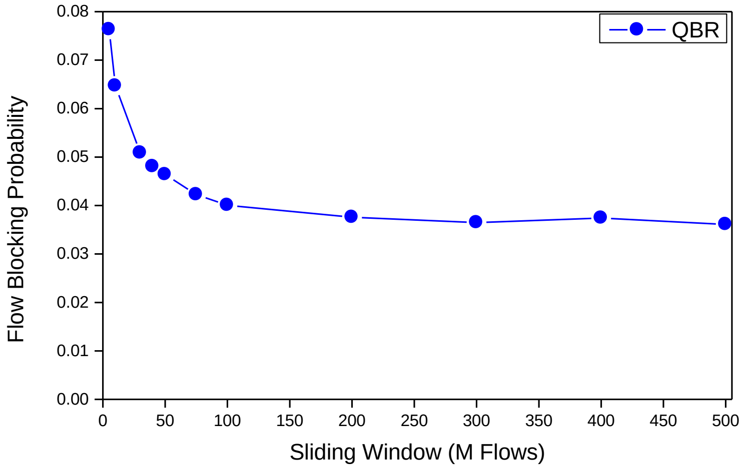


Figure 4.8 Choices of window size (M Connection Requests) in Random 32

4.4.4.7 Time complexity

Most global QoS routing algorithms perform variants of Dijkstra's algorithm on the resulting graph. This takes at least O ($N \log N+L$) time, where N is the size of the network (nodes) and L is the number of links (edges) in the network. On the other hand, the CBR, ABR and QBR schemes use a simple method that involves path selection among the set of candidate paths R, in which the time complexity requires O ($|R|$) operations. CBR needs to update the blocking probability which requires O ($|1|$). Similarly, QBR needs to update the bottleneck residual bandwidth statistics and ABR needs to update acceptance rate statistics, which also both require a constant time O (1).

In fact, QoS routing is scalable when the overhead increases linearly as the number of nodes increases. In global QoS routing each node requires a view of the status of the whole network throughout link state updates. On the other hand, in Localised QoS routing only source nodes need to monitor statistics about the candidate path set and send a flow to the most feasible path. Moreover, Localised algorithms do not need the global state to be maintained at each router to keep the link state updated as this imposes routing overheads.

4.5 Summary

In this chapter, different from previously used Localised algorithms, which indirectly characterise the quality of the path, a Localised acceptance rate based QoS routing scheme (ABR) using the acceptance rate for routing using an average acceptance rate to a Localised quality based QoS routing scheme (QBR) that makes routing decisions using bandwidth statistics collected locally. A simulation model was developed to evaluate the proposed performance in realistic scenarios.

Performances of these algorithms were compared against the CBR algorithm and it was demonstrated, through extensive simulations using different types of network topologies and under a wide range of traffic loads, that these algorithms perform better than the CBR algorithm in all situations. Simulation results showed that ABR and QBR gave a comparable performance with low communication overhead.

Only a portion of the results were observed in this chapter the findings were published in a research paper [118], but for the purpose of this thesis, many details had to be added and corrected even for the mentioned parts.

Chapter 5

A new Localised Network Based Routing Algorithm

5.1 Introduction

The constantly changing nature of the Internet presents a challenge in evaluating routing behaviour in large-scale networks. Various studies have evaluated the performance of different routing algorithms on simulated networks with various topologies, such as ISP [48] [72], random topologies [2], and others [49]. However, due to the characteristics and decentralized nature of the Internet it makes it hard to define a typical topology [65] [66]. In view of the fact that routing algorithms are network layer entities and the varying performance of any routing algorithm depends on the underlying network topology, in the efficient design and assessment of the performance of Localised routing algorithms it is necessary to use different network topologies which vary in their characteristics and topology parameters. These differences between topologies have a significant influence on how well the routing algorithms behave.

It has been shown from the previous chapter that Localised routing algorithms discriminate against alternative paths which prefer the shortest paths. These minimum hop algorithms usually outperform algorithms that do not take path length into consideration [1] [64] [73]. Biasing towards short paths is particularly attractive in large-scale networks, since path length is a relatively stable metric compared with link delay [97]. With this in mind, It is concentrated to shed some light on how Localised QoS routing algorithms can be affected by the level of network connectivity.

In this chapter a new network parameter is developed that can be used to predict which network topology gives better performance on the quality of Localised routing algorithms, taking path length into consideration. A simple method is proposed using this parameter which can be rewired to introduce increasing the connections that the network accepts. Simulations of random and complex networks used to show that the performance is significantly affected by the level of connectivity.

5.2 Some Basic Network Features

The ubiquitous nature of complex networks has been recently studied in many fields in science [99] [100] [101], such as the Internet, the power grid and the World Wide Web (WWW). These have naturally been concerned with the network structure and connectivity constraints. This section provides details of the main concepts which have played key roles in recent study.

5.2.1 Average Path Length

The average path length $\bar{h}$ is defined as the number of links in the shortest path distance between two nodes, averaged over all pairs of nodes.

$$\bar{h} = \frac{2}{N(N-1)} \sum_{i=1}^{N} v_{ij} \qquad \text{..................} \quad (5.1)$$

where v_{ij} is the distance between node i and node j, and N is the number of nodes. The diameter D of a network therefore is the maximum distance between any pair of nodes in the topology.

The average path length $\bar{h}$ determines the effective size of the network; the average path length is relatively small in most realistic networks, implied by the name *small world networks.*

5.2.2 Clustering Coefficient

The clustering coefficient $\overline{C}$ is the average fraction of pairs of neighbours of a node that are also neighbours to each other. Suppose that a node i has k adjacent nodes to the maximum possible number which is given, at most $k_i\left(k_i-1\right)/2$ links can exist between them. Let C_i denote the fraction of allowable links that actually exist. $\overline{C}(i)$ is then the clustering coefficient over all node i.

$$\overline{C}(i)=\frac{2C_i}{k_i(k_i-1)} \qquad \text{………………………..} \quad (5.2)$$

Every node in the topology has a clustering coefficient associated with it. The clustering coefficient of whole topology is the average of $\overline{C}(i)$ over all node coefficients.

$$\overline{C}=\sum_{i=1}^{N}\frac{\overline{C}_i(i)}{N} \qquad \text{……………………… …..} \quad (5.3)$$

where N is the number of nodes.

5.2.3 Degree Distribution

The most important characteristic of a single node is its degree. The degree d_u of a node u is defined by the total number of its connections $d_u=2L$ where L is the number of links. Thus, the larger the degree the more important the links of the node are in the network. The average of $\langle d \rangle$ over the whole topology is called average node degree of the network, and can be written as $\langle d \rangle = \frac{2L}{N}$ where L is the number of links and N is the number of nodes.

The spread of node degrees over a network is characterized by the degree distribution function $P\langle d \rangle$, which is the probability that a randomly selected node has exactly d links over the whole network. A prominent common feature of random topologies is that the connectivity distribution of a network peaks at an average value and decays exponentially. The degree distribution can be approximated to a binomial degree distribution.

$$p\langle d \rangle = \binom{N-1}{d} P^{d} (1-p)^{N-1-p} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5.4)$$

For large N it can be replaced by a Poisson distribution as:

$$p\langle d \rangle \cong e^{(-PN)} \frac{(PN)^{d}}{d!} = e^{-\langle d \rangle} \frac{\langle d \rangle^{d}}{d!} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5.5)$$

where $\langle d \rangle$ is the average node degree of topology.

Although the previous metrics are major factors used to measure the topology used in many realistic networks, they only measure the node clustering structure and do not possess the capability to capture the global clustering structure. Clustering coefficients compute the clustering with regard to how two hop neighbours are connected. How are all nodes correlated? Similarly, the degree distribution is too limited to capture global properties of Internet topologies; topologies with the same degree distribution may differ significantly [109]. The average path length computes the distance between source destination pairs. However, the topology metrics based on average values do not reflect balance clustering through all nodes in the topology.

5.3 Small World Networks

Recent work on Internet topology characterisation and complex networks has been related to small world models featuring distinct elements from the early Erdős and Rényi (ER) random network model [102] and Waxman’s model, which extends the ER model by factoring the distance between nodes [71].

5.3.1 Small World Networks

Watts and Strogatz [103] introduced the concept of a small world network in terms of transition from a regular topology to a random topology. When considering the high level of clustering of a regular topology (lattice), it starts with a ring lattice with N nodes in which every node is connected to L neighbour, thus showing a large clustering coefficient with a small average path length. It goes with a randomly rewiring links in the network with probability p in such a way that the transition between order $p = 0$ and randomness $p = 1$.

However, the major limitation of the algorithm is that it can be viewed as homogeneous and the degree distribution observes the same binomial distribution used for random topologies. Although the authors in [110] observed that the Internet topology has small world behaviour, when recent topology models were evaluated none of those modelled resulted in high clustering coefficients for the Internet [109].

5.3.2 Scale-Free Networks

Barabási and Albert (BA) [104] suggested that the complex networks are not exponential networks, they are in fact *scale-free networks* and their degree distribution has a power law [105]. Faloutsos observed the same power law for the Internet [106]. A power law does not observe a peak value at its mean value. Rather, it starts at the maximum value and then decreases with a characteristic exponent γ all the way to infinity.

$$p\langle d\rangle \approx d^{-\gamma} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5.6)$$

This is why they are called scale-free networks, as they have no characteristic scale to define degree distribution properly. Recent studies of degree distributions on the Internet, WWW and cellular networks (among others) can fit the above equation with γ values between 2 and 3, depending on the particular network [108].

On the other hand, the number of links for most nodes is small, while only a few nodes have a higher number of links. This is because scale-free networks grow through the addition of new nodes that are preferentially connected to the existing nodes with a high degree of connectivity. This high connectivity nodes act as *highly connected hubs*, hubs are responsible for the small world phenomena. Hubs are commonly referred to, in the large-scale Internet, as autonomous systems (AS) [115].

However, there is no analytical justification formula for the average path length or clustering coefficient for the scale-free model and the model, in general, fails to produce the high levels of clustering seen in real networks. A different research direction considers the tradeoffs between performance and physical topology constraints and suggests that physical topology should not be a power law to enjoy better performance [107].

Although, various topologies manifest obvious small world properties, it has been shown that small world networks and scale-free topologies are rewiring randomly and they are in fact resulting in very sparsely connected networks. However, their topologies are not balancing the connectivity between nodes and it is inadequate for the performance of Localised routing algorithms.

5.4 A New Clustering Metric

Previous studies in the context of global QoS routing algorithms have tended to use average path length and the average node degree [2] [73] [98] to measure the level of connectivity. Clustering structures have a significant impact on the performance of various routing protocols [111]. Further topology parameters can help the analysis of network traffic, congestion and critical network issues [112]. However, Localised QoS routing algorithms need a suitable topology that balances the distance between any pair of nodes in the topology to increase the flexibility of a large number of possible candidate paths, which would reduce the blocking probability. For this reason it is desirable to design a new topology metric that accurately captures the balance level of connectivity. A balance clustering metric (BCM) is developed which is a practical clustering accuracy metric. The basic idea of BCM involves calculating the distances between any pair of nodes and then computing a s*tandard deviation path length* which subjectively specifies how tightly the nodes are clustered throughout the topology. To the best of our knowledge the proposed metric has not been analysed before, whether in the context of routing algorithms or topology based models.

Let V be a set of topology nodes and L be the set of links in a topology $G(V,L)$, the BCM metric using path length for each pair of nodes Λ_i, let $\widehat{M}$ be the path matrix which stores the number of hops along the direct path between source and destination pairs, $\widehat{M}$ is the shortest path length among all node pairs and the diagonal elements of $\widehat{M}$ are zero. The diagonal elements are all zero $M_{ii} = 0$, and off-diagonals contain the number of hops (distance) along the path connecting each pair of nodes $i \neq j$ is v_i.

Therefore the BCM metric is calculated using the *standard deviation of the row distances* as follows:

$$BCM = \sqrt{\frac{\sum_{i=1}^{N}\left(\Lambda_i - \overline{m}\right)^2}{N-1}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (5.7)$$

Where, Λ_i is the sum of rows or columns in the path matrix and $\overline{m}$ is the arithmetic mean. The sum of row distances for each node is calculated by:

$$\Lambda_i = \sum_{i=1}^{N} v_i \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.8)$$

The arithmetic mean is $\overline{m} = \sum_{i=1}^{N} \Lambda_i / N$, the BCM metric of the whole topology is:

$$BCM = \sqrt{\frac{\sum_{i=1}^{N}\left(\sum_{i=1}^{N} v_i - \left(\sum_{i=1}^{N} \Lambda_i \Big/ N\right)\right)^2}{N-1}} \quad \text{........................} \quad (5.9)$$

where N is the number of nodes.

The BCM metric can well reflect the significance of the structure in the topology. The smaller the BCM the more significant the structure, if the BCM=0 this means the distance between each source-destination pair is the same; this is considered to be a balanced topology.

the average path length ($\overline{h}$) is calculated by summing the non zero elements of the path matrix and dividing by the number of non zero elements.

$\overline{h} = T / N(N-1)$,where T is the sum of off-diagonal elements and N is the number of nodes.

the degree and clustering coefficients of topology characteristics is obtained using its adjacency matrix $\widehat{A}$. Any network topology G with N nodes can be represented by its adjacency matrix $\widehat{A}$ with $N \times N$ elements A_{ij}, whose value is $A_{ij} = Aji = 1$ if nodes i and j are connected and 0 otherwise.

The degree of the node is $k_i = \sum_j A_{ij}$, the total degree of the topology is the doubled total number of links L $K = 2L = \sum_{i=1, j=1}^{N} A_{ij} = \sum_{i=1}^{N} (\widehat{A}^2)_{ii}$

The total number of loops of length 3 in the topology is:

$$N_3 = \frac{1}{6} \sum_{i=1}^{N} (\widehat{A}^2)_{ii}$$

The total number of connected triples of nodes in the topology is:

$$T_r = \frac{1}{2} \sum_{\substack{i=1, j=1 \\ i \neq j}}^{N} (\widehat{A}^2)_{ij} = \frac{1}{2} \sum_{i=1}^{N} k_i (k_i - 1)$$

The clustering coefficient of the topology is:

$$C(i) = \frac{1}{9} \frac{\sum_{i=1}^{N} (\widehat{A}^3)_{ii}}{\sum_{\substack{i=1, j=1 \\ i \neq j}}^{N} (\widehat{A}^2)_{ij}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.10)$$

More information for indicating the clustering coefficient relations can be found in [113].

For example, a simple topology depicted in Figure 5.1 is evaluated, which the number of nodes N=6 and the number of links L=9.

The diagonal elements are all zero and off diagonals contain the number of hops along the path connecting each pair of nodes. In fact the diameter of this topology is 2 hops; therefore the largest value for any element of L can be 2.

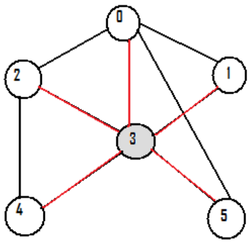


Figure 5.1 Illustration of BCM Metric

The average path length ($\bar{h}$) is 1.4 the clustering coefficient is 0.9133 and the balance clustering metric (BCM) is therefore is 1.265

	v_0	v_1	v_2	v_3	v_4	v_5		Λ_i		BCM
v_0	0	1	1	1	2	1		6		
v_1	1	0	2	1	1	2		7		
v_2	1	2	0	1	2	2		8		
v_3	1	1	1	0	1	1	=	5	=	1.265
v_4	2	1	2	1	0	2		8		
v_5	1	2	2	1	2	0		8		

Table 5.1 Illustration of BCM Metric

5.5 Localised BCM Network Model

A simple method is developed using a BCM metric that can be rewired to increase the connections that the network accepts. Thus, the topology is assumed that is static in the sense that although links can be rearranged, the number of nodes is fixed throughout the forming process. However, our goal is not to provide a thorough evaluation of topology model design, rather the aim is to provide an accurate topology structure that maintains best performance for Localised routing algorithms.

It is necessary to develop a simple method for network topology in which the level of connectivity can be adjusted in a controlled manner. This method incorporates the addition of new links, the removal of some existing links and rewiring without altering the number of nodes or links in the topology. The path matrix form is examined in which the elements of path matrix $\widehat{M}$ are equal to the distance separating nods or 0 if no path exists.

Thus, starting from the initial topology structure, it is necessary to monitor changes of topology by removing links for lower distance between existing node pairs and reconnecting to larger node distance pairs in agreement with the level of connectivity: removing the link $\min \Lambda_i$ to $\min v_i$ connected nodes and adding a link to $\max \Lambda_i$ to $\max v_i$ node pairs.

This method is quite natural, since it is possible to make several rewiring changes to decrease the BCM metric and then decrease the average path length significantly. On the other hand, several rewired links will not crucially change the local clustering property of the network. The path matrix of the example described in the previous Section is examined to illustrate BCM network method, the number of nodes N=6, the number of links L=9, the clustering coefficient is 0.91 and $BCM = 1.265$, as shown in the Table 5.2:

	v_0	v_1	v_2	v_3	v_4	v_5		Λ_i		BCM
v_0	0	1	1	1	2	1		6		
v_1	1	0	2	1	1	2		7		
v_2	1	2	0	1	2	2		8		
v_3	1	1	1	0	1	1	=	5	=	1.265
v_4	2	1	2	1	0	2		8		
v_5	1	2	2	1	2	0		8		

Table 5.2 Path Matrix of original Topology

The link between nodes $v_3 \rightarrow v_0$ is removed which is the lower Λ_i and add a link between nodes $v_5 \rightarrow v_4$ which is the larger Λ_i.

The rewiring process is repeated according to the level of connectivity, the following example demonstrates three rewiring changes and the BCM value has decreased from 1.265 to 0. The path matrix and topology representation is then modified as follows:

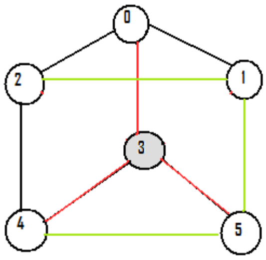


Figure 5.2 Illustration of BCM Topology

	v_0	v_1	v_2	v_3	v_4	v_5		Λ_i		BCM
v_0	0	1	1	1	2	2		7		
v_1	1	0	1	2	1	2		7		
v_2	1	1	0	2	2	1		7		
v_3	1	2	2	0	1	1	=	7	=	0
v_4	2	1	2	1	0	1		7		
v_5	2	2	1	1	1	0		7		

Table 5.3 Path Matrix of BCM Topology

It is noted that the average path length is 1.4, the clustering coefficient is decreased to 0.33 and the BCM metric decreased to 0.

5.6 Performance Evaluation

The performance of the proposed method is evaluated using the Localised QBR scheme and the Localised CBR scheme. In the following, the simulation model is described first and then compare the performance of the QBR and CBR schemes using the performance metrics defined in the previous chapters.

5.6.1 Simulation Model

Extensive simulations are conducted and implemented under OMNeT++ [60] [61] in order to test the performance of the Localised QoS routing algorithms proposed in the previous chapter, according to the level of network connectivity. Consequently, the simulation experiments considered a range of topologies with similarities and differences in their important parameters, such as average path length, node degree and diameter, including BCM metric. This study is comment on similarities and differences between the trends in each topology.

Random topologies are considered with relatively different levels of connectivity to evaluate the effects of having multiple shortest candidate paths between pairs of nodes. In addition, the well-known ISP topology is considered which has been classified as a sparsely connected topology. The random topologies were generated using the BRITE generator [70] using Waxman’s model [71]. The scale-free topology is considered which was generated on top of the BRITE generator [70] using the Barabási-Elbert model [100].

5.6.2 Traffic Generation

Traffic engineering assumptions and parameters is stated in Section 4.4.2 including the capacity, flow arrival distribution QoS requested distribution, offered network load formula and the parameters has been used in the simulation for CBR algorithm.

5.6.3 Performance Metrics

The performance metrics is stated in Section 4.4.3 including flow blocking probability and bandwidth blocking probability formulas and definitions.

5.6.4 Simulation Results

5.6.4.1 Performance Prediction of Localised Routing Algorithms

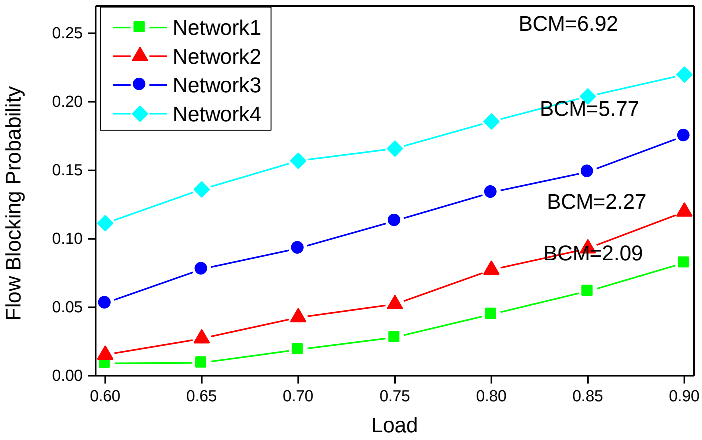
The Localised QoS routing algorithms depend on the underlying network topology. To study the effects of topology on the performance, four random topologies are evaluated with similar size n = 18 and number of links L=58 under the same traffic load using the topology metric described in Section 5.4.

The main parameters and characteristics of the configurations are listed in Table 5.4.

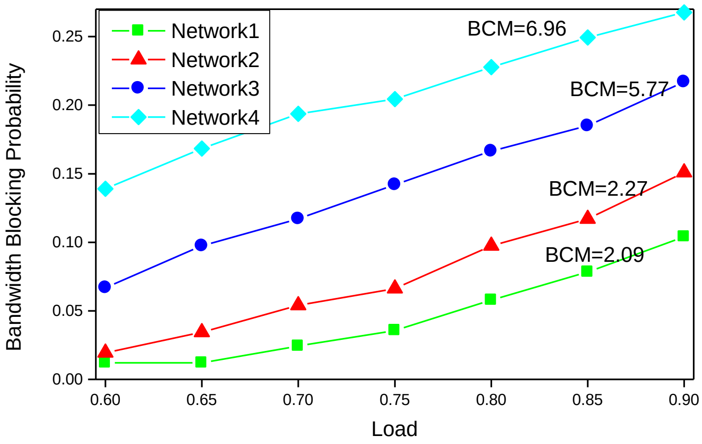
Name	NODES	LINKS	Avg. degree	Avg. path length	BCM Metric
Network1	18	58	3.22	2.32	2.09
Network2	18	58	3.22	2.346	2.27
Network3	18	58	3.22	2.43	5.77
Network4	18	58	3.22	2.5	6..92

Table 5.4 Characteristics of random topologies

It can be shown from topologies parameters that the balance clustering metric BCM is increasing based on the level of connectivity. A lower BCM typically implies a dense topology with balanced path lengths and more flexibility in selecting routes, whereas a larger BCM implies lower connectivity. The differences in the BCM metric shown in Figure 5.3 reveal a significant influence regarding how well Localised algorithms perform.



(a) CBR (Flow Blocking Probability)

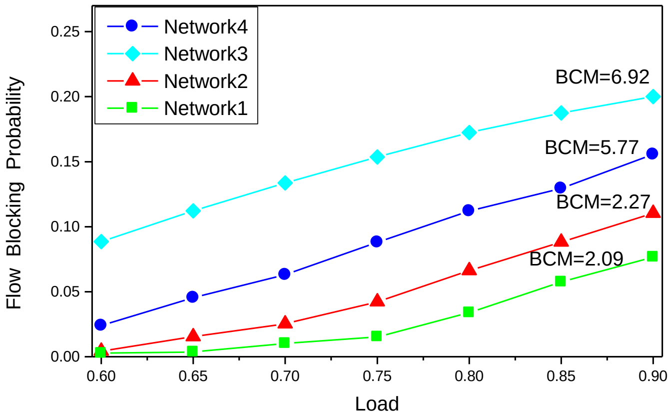


(b) CBR (Bandwidth Blocking Probability)

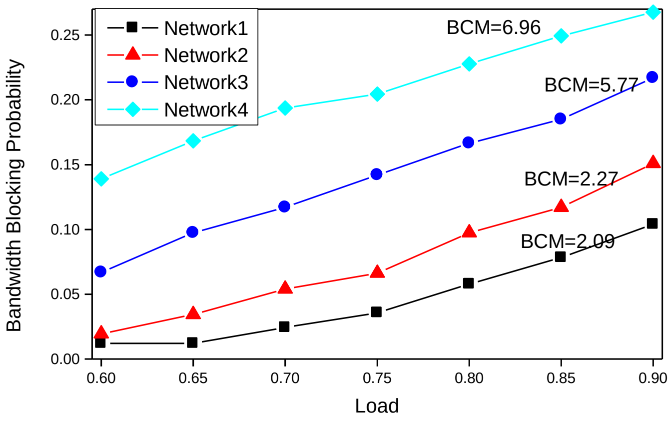
Figure 5.3 Comparison of random topologies of CBR scheme using flow and bandwidth blocking probabilities

Figure 5.3(a) and Figure 5.3(b) show the comparison between topologies using Localised CBR scheme in terms of flow blocking probability and bandwidth blocking probability. The performance metrics are plotted as a function of offered load. The offered load is increased by changing the mean holding time using the formula described in Section 4.4.2. The performance of Localised CBR strongly depends on the network topology measured by the value of the BCM metric; the Localised routing algorithms perform well for dense topologies. The reason behind that, dense random topologies have flexibility of a large number of possible candidate paths which reduces the blocking probability.

However, the performance for the random topology with a lower connectivity degrades, which is more likely when most pairs of nodes have a larger path length of the network.



(a) QBR (Flow Blocking Probability)



(b) QBR (Bandwidth Blocking Probability)

Figure 5.4 Comparison of random topologies of QBR scheme using flow and bandwidth blocking probabilities

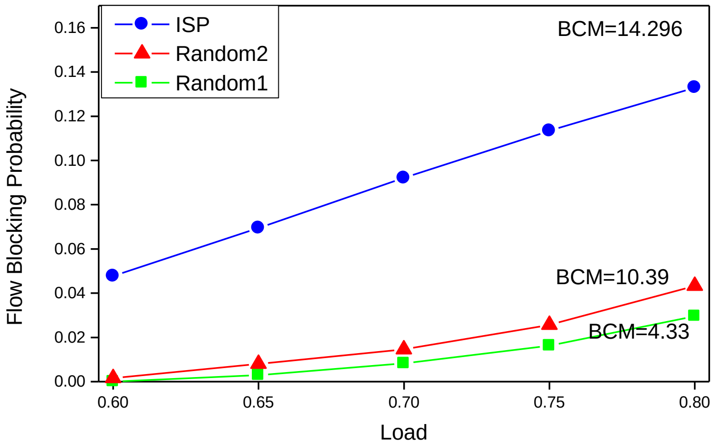
Similarly, Figure 5.4 compares the same topologies using both performance metrics for the Localised QBR algorithm, the performance of the QBR algorithm depends on the level of connectivity.

The performance of sparse ISP topology is evaluated, as depicted in Table 5.2. However, the characteristic of ISP topology has relatively lower connectivity compared to random topologies. Table 5.5 lists the main parameters of the proposed topologies.

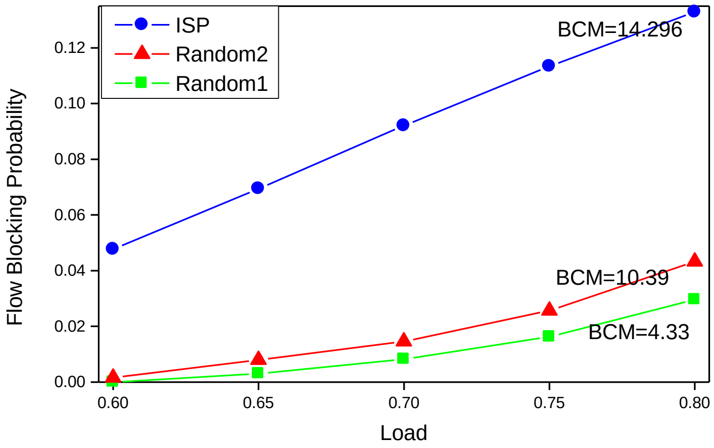
Name	Nodes	Links	Avg. degree	Avg. path length	BCM
ISP	32	108	3.375	3.177	14.296
Random1	32	122	3.8125	2.494	4.329
Random2	32	122	*3.8125*	2.416	10.388

Table 5.5 Characteristics of random and ISP topologies

Figure 5.5 shows the comparison between ISP and random topologies. It can be seen that the performance degrades as the value of the BCM metric increases. Figure 5.5 shows that Localised routing algorithms usually discriminate against alternative paths in which the algorithms prefer the shortest paths. It has been shown that the Localised routing algorithms typically perform well in dense networks (i.e. Random 32 topology) which have the flexibility of a large number of routes between each source and destination.



(a) CBR



(b) QBR

Figure 5.5 Comparison of random and ISP topologies using flow blocking probability

On the other hand, these algorithms are hard to use scarce resources in sparsely connected ISP topologies as they consume more resources to accommodate future flows.

More generally, due to the growth of the Internet and increasing demand for predictable performance, the simulation results observed an ability to predict topology. Using the BCM metric it is possible to predict, between topologies, which are likely to give better performance.

5.6.4.2 Network Based Localised Routing Model

Figure 5.6 shows simulation results of the ISP topology which continuously adds links and removes existing links to the topology using the method described in Section 5.5 until a rich connectivity is achieved.

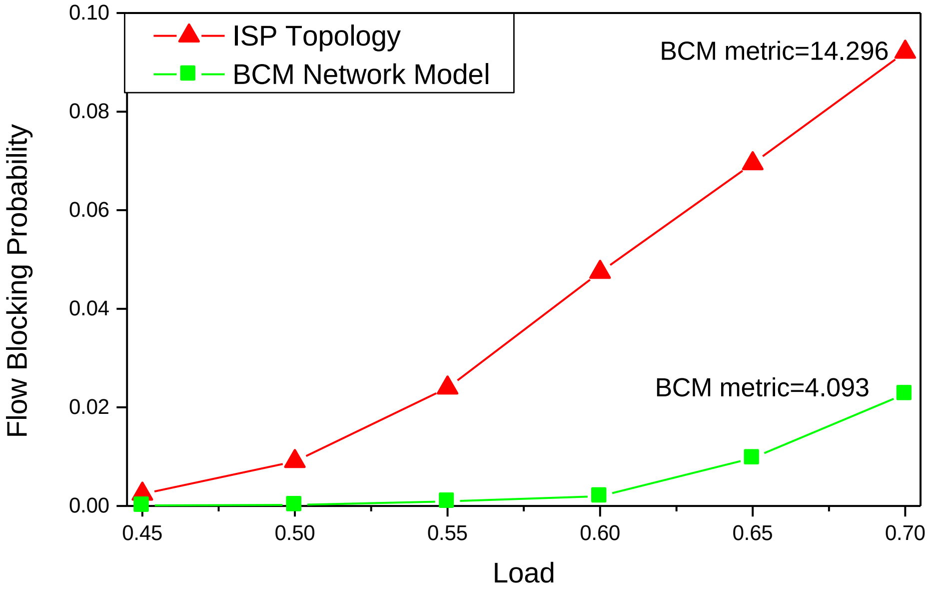


Figure 5.6 Comparison of the BCM network model with ISP topology

Figure 5.6 compares the performance of the ISP topology and BCM topology. It can be observed that the BCM reduce blocking probabilities as well as increase the number of candidate paths for each node pair. Using larger paths in sparse topologies tends to consume more resources for future flows causing performance degradation. In this form it is of negligible importance when load is low, as there are still sufficient resources to route future flows.

Figure 5.7 compares the performance of the QBR algorithm under the scale-free topology and BCM topology, the performance is increased by changing the topology to BCM topology.

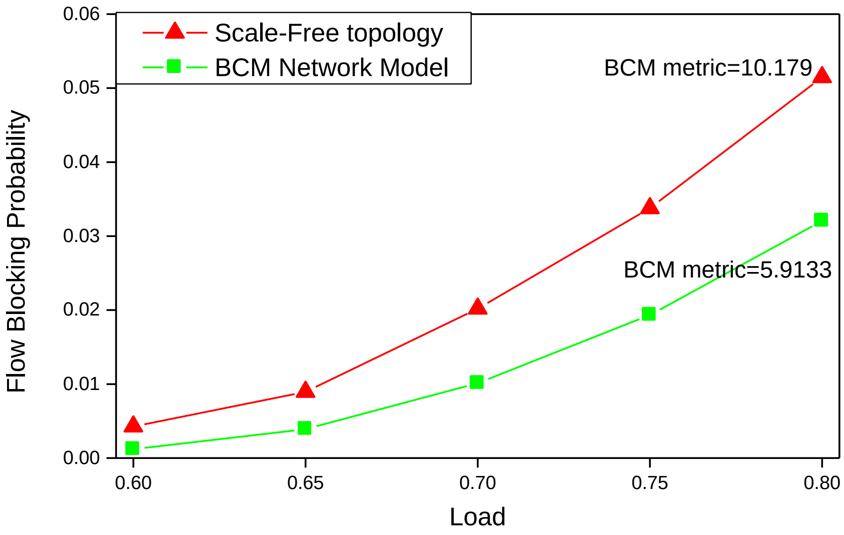
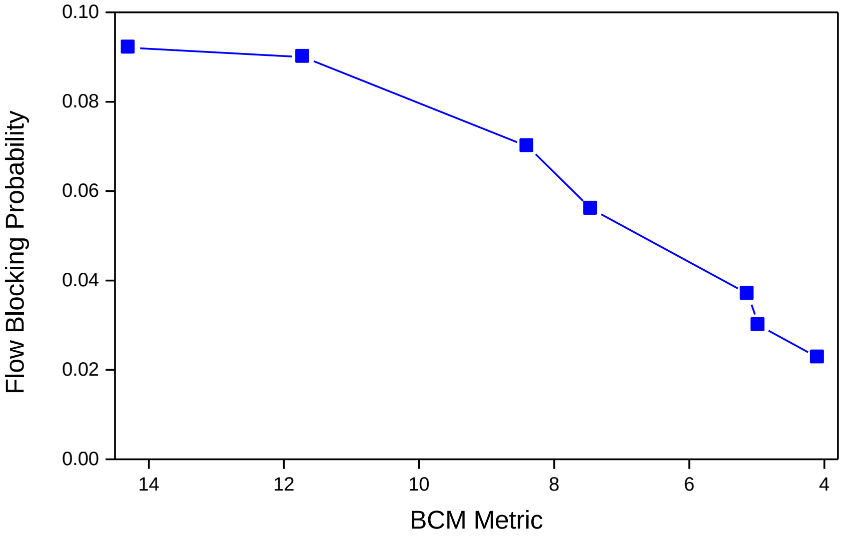
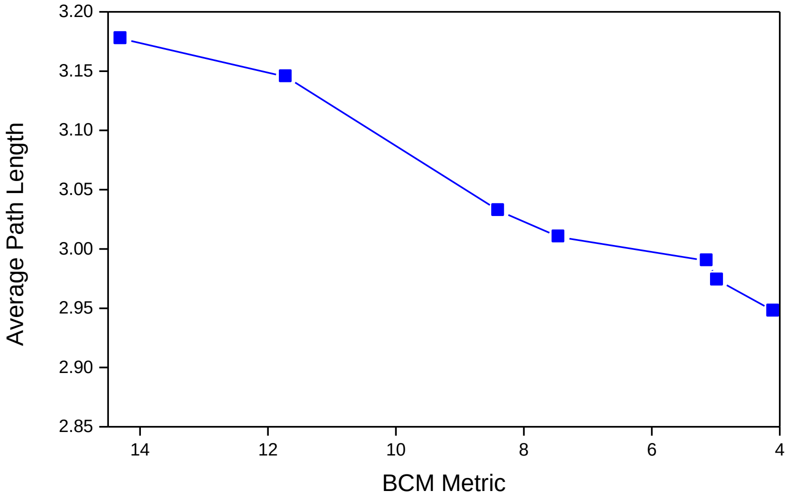


Figure 5.7 Comparison of the BCM topology with scale-free topology



(a) Flow Blocking Probability



(b) Average Path Length

Figure 5.8 Flow blocking probability and average path length

Figure 5.8 (a) observed a decreasing blocking probability as a function of decreasing the BCM metric. Figure 5.8(b) observed a decrease in the average path length in the BCM topology which significantly satisfies the small world property which maintaining the same network sized of the original network.

5.6.4.3 Varying Non-Uniform Traffic

The destination nodes in the simulation experiments have been selected uniformly. However, in realistic networks the source nodes may receive more traffic demands, especially in the case of the communications in the sub-networks, which usually receive more demands than the communications across sub-networks.

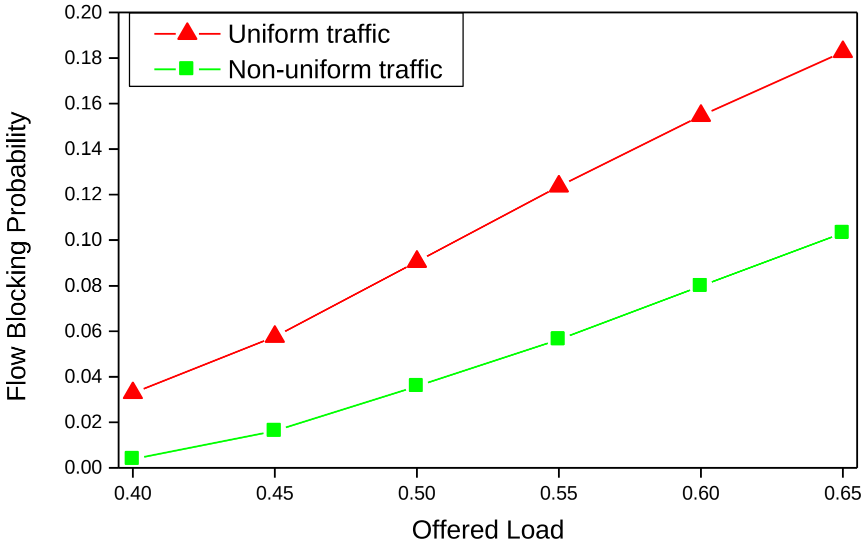


Figure 5.9 Impact of varying non-uniform traffic

The authors in [114] emphasized that the global uniform end-to-end IP QoS solution is not realistic. For these reasons, non-uniform traffic is modelled under the scale-free topology and it has been virtually divided into two sub-networks. The BCM metric of the sub-network1 is 4.22, the BCM metric of the sub-network2 is 4.5 and the BCM for the network topology is 8.72.

The non uniform traffic is varied across the scale-free topology; the traffic demands routed inside the sub-network are three times higher than traffic routed between sub-networks.

In Figure 5.9 flow blocking probability is plotted against different load conditions using uniform and non-uniform traffic for scale-free topology. It can be noticed that the Localised routing algorithms perform extremely well under non-uniform traffic compared with uniform traffic. Hence, by formulating the correlated traffic in the building blocks of realistic topologies using the BCM metric, superior performance is effectively obtained by reducing the blocking probability.

5.7 Summary

In this chapter a new clustering metric is developed based on Localised QoS routing taking path length into consideration. The BCM metric showed to be a good indicator and can be used to predict which network topologies give better performance using the same network features. The BCM network model is developed using a BCM metric that rewired to introduce increasing the performance. It is found that this model have small characteristic path length under random and complex network topologies. Localised QoS routing algorithms typically perform well for highly connected networks where they are likely to be able to balance the load over the set of minimum hop candidate paths. The performance is assessed under a spectrum of topologies with various connectivity levels to model the diversity seen in current realistic networks.

However, due to Localised algorithms being difficult to route for scarce resources over sparse topologies, it is generally suggested that the Localised algorithms should distribute the load according to connectivity, in order to obtain a comparable performance and then avoid congestion, this is the major work undertaken in the following chapter.

Only a portion of the results were observed in this chapter; these were published in a research paper [119], but many details had to be added for this chapter.

Chapter 6

A new Localised Call Admission Control Algorithm

6.1 Introduction

Future large-scale networks such as the Internet will carry a wide spectrum of new multimedia applications, such as VoIP, video on demand, etc.

A key characteristic of these applications is that they require Quality of service guarantees. In particular create a connection and reserve it on the path between source and destination pairs. Call admission control algorithms are needed since network resources are limited. The call admission control is considered one of the important parameters that can affect the performance of QoS routing algorithms [117]. Call admission control is desirable to reject a request even when a feasible path has been found.

However, the previous Localised algorithms were only path selection routing algorithms and this leads to them accepting every incoming flow that can be physically accommodated. It is suggested that Localised QoS routing algorithms should be supplemented by a call admission control in order to maximize the connection requests that the network accepts, distribute the load throughout the network and thus reduce the overhead by minimizing the signalling effort. Simulations of various network topologies are used to illustrate the performance of the algorithm using various ranges of traffic loads.

6.2 Higher-Level Call Admission Control

This study distinguishes between link admission control and higher-level call admission control. Link 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 [7], efficiency of QoS routing may require an additional layer of admission control. Higher-level call 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 [7] [117].

6.3 The Congestion Avoidance Routing Algorithm

The increasing demand for real-time multimedia applications has motivated researchers to develop routing algorithms to accommodate resources, such as the QoS for the Internet. It has been shown from the previous chapters that Localised routing algorithms discriminate against alternative paths which prefer short paths.

With the explosive growth of Internet traffic, these characteristics of short path preferences lead to major drawbacks; by only using the links belonging to the short paths they become heavily loaded and this leads to *congestion* on these short paths, while the longer paths are under-loaded and may not be used at all [46] [116]. This makes unbalanced traffic distribution, and the resources are not used efficiently. Moreover, the previous chapter results observed that Localised QoS routing highly depends on network topology, and it has been shown that the above situation happens usually for sparse topologies with low connectivity, where the Localised routing algorithms show poor performance.

On the other hand, Localised QoS routing algorithms perform routing decisions based on information updated from path request to path request, since future connection requests will be routed via congestion links until the whole cycle will repeat. In this chapter a new Localised congestion avoidance routing (CAR) algorithm is introduced to enable decisions of routing to be made to avoid congestion for each connection request; selection of disjoint paths between source and destination will tend to show progress in reducing the overall blocking probability.

The performance of the algorithm is evaluated using extensive simulations. Simulation results show that the CAR algorithm outperforms the credit based routing (CBR) [58] algorithm and quality based routing (QBR) algorithm under various ranges of traffic loads.

The mechanism presented differs from previous Localised QoS routing algorithms in the following ways:

- Previous Localised routing algorithms select a candidate path based on information (i.e. blocking probability, bandwidth) updated from path request to path request. In contrast, the CAR algorithm performs routing at the source node based on congestion state in each connection request.
- The CAR algorithm performs routing by selection of disjoint paths away from candidate paths that consist of congestion links.
- The CAR algorithm routes traffic over sparse topologies which would make the behaviour of Localised routing algorithms less dependent on network topology and would to enable use of scarce resources efficiently.

The congestion avoidance routing (CAR) algorithm is a source routing algorithm, where source routing makes the 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.

The pseudo code for the CAR algorithm is given in Figure 6.1, as follows:
Localised congestion avoidance 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 congested link and a failure message is propagated back to the source node. The algorithm starts to select disjoint paths between the source and destination. If any of the future flow arrivals go through the congested candidate path the algorithm replaces the path with a second higher quality path (line 5) 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.

In terms of the path request basis, upon the flow being accepted along the path (line 8), the residual bandwidth is calculated, if its value is greater than that requested it is updated by one (line 10), or if it is smaller, it is updated by 0.5, which estimates and reflects the quality of the path (line 13). On the other hand, upon the flow being rejected the quality is updated to zero (line 18). Upon flow being rejected the algorithm stores the congested link for the next arrival of connection requests (line 16).

```
PROCEDURE CAR( )
Initialisation ( )

Set P.Avg=1,$\forall$ P$\in$R

CAR( )

1. if P.Avg=0 $\forall$ P$\in$R

2. Set P.Avg=1, $\forall$ P$\in$R
```

```
Set $P=\max\{P.Avg: P \in Rmin\}$.

Connection Request Update:

If flow routed through congestion path

    Change path P=second cand. Path.

else

    Route flow along path P

Path Request Update:

if flow accepted{

    if(ResidualBanndwidth>=Requested Bandwidth){

        Set the value=1;
        Calculate (P.Avg,1)}

    else if(ResidualBanndwidth<RequestedBandwidth){

        Set the value=0.5;
        Calculate (P.Avg, 0.5)}}
else

    Set Congested link
    Set the value=0;

    Calculate (P.Avg,0}

END PROCEDURE
```

Figure 6.1 CAR pseudo code

The CAR algorithm make transition from static to dynamic update information and uses selection of disjoint paths and recalculation of the set of candidate paths. A further study of these path selection methods can be found in [123] [121]. However, the main focus of this chapter will on integrated of call admission control and Localised QoS routing algorithms and they are still only path selection routing algorithms.

6.4 Localised Call Admission Control Algorithm

The mechanism presented in this chapter differs from previous Localised QoS routing algorithms in the following ways:

- Previous Localised routing algorithms were only path selection routing algorithms. The Localised call admission control with Localised routing algorithm is developed. To the best of our knowledge the call admission control algorithm has not been studied in the context of Localised routing algorithms.
- Previous Localised routing algorithms route a connection if there is a path with sufficient bandwidth. Unfortunately, this approach can lead to congestion. It will accept a connection even if that connection can only be accommodated along an excessively long path that might be more efficiently used by some future connections.
- The resources are reserved using routing algorithms as well as call admission control algorithms. The call admission control algorithm decides which connection should be accepted and which should be rejected. The routing decides the path selection used by the algorithm.

The Proposed Algorithm

A new call admission control algorithm (CAC) was used with Localised 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 Localised PSR, CBR, QBR and CAR Localised 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 Localised algorithms since call admission control algorithms proposed to manage the incoming flows.

The pseudo code for the CAC algorithm as given in Figure 6.2, as follows:

```
Initialize ( )
Set $P.Avg=1, \forall P \in R$
CAC( )
1. if $P.Avg=0 \ \forall P \in R$
2. Set $P.Avg=1, \ \forall P \in R$
3. Set $P=\max\{P.Avg: P \in Rmin\}$.
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 6.2 CAC algorithm pseudo code

Localised 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.

6.5 Performance Evaluation

We now evaluate the performance of the proposed Localised algorithms; the CAR and CAC algorithms are compared against the Localised CBR algorithm and Localised QBR algorithm. PSR and global QoS routing algorithms for comparison since Localised CBR has been shown to outperform both PSR and the global QoS routing (WSP) algorithm in all situations [2] [55].

6.5.1 Simulation Setup

We implemented the Localised QoS routing scheme based on the discrete-event simulator OMNeT++ [60] [61] and conducted extensive simulations to test the performance. A variety of QoS routing schemes [23] [45] [64] [69] have been proposed to make routing decisions at the flow level. The predetermined Localised PSR and CBR algorithms perform routing decisions in the simulation at the flow level. By performing simulation at the flow level it is possible to simulate the performance of thousands of flows in an efficient manner, and consequently, the *objectives and procedures* differ from the packet level. Hence, it is only needed to perform flow level simulation to study the performance of the CAR algorithm.

Due to the fact that performance of the QoS routing algorithms may in fact vary with underlying network topologies [73], our simulation experiments consider different types of network topologies. A familiar ISP topology is used in the simulation, which is widely used in different QoS routing algorithm studies [68] [69] [72]. In addition, random and scale-free topologies are investigated. The random and scale-free topologies were generated using the BRITE generator [70]. Table 6.1 lists the most important characteristics of the topologies used in the simulation.

Topology	Nodes	Links	Avg. degree	Avg. Path Length	BCM Metric
ISP	32	108	3.375	3.177	14.296
RANDOM 32	32	122	3.812	2.416	10.388
SCALE-FREE	32	122	3.812	2.4274	10.179

Table 6.1 Topologies and their characteristics

6.5.2 Traffic Generation

Traffic engineering assumptions and parameters is stated in Section 4.4.2 including the capacity, flow arrival distribution QoS requested distribution, offered network load formula and the parameters has been used in the simulation for CBR algorithm.

6.5.3 Performance Metrics

The performance metrics is stated in Section 4.4.3 including flow blocking probability and bandwidth blocking probability formulas and definitions.

In our fairness calculation, the well-known Jain’s fairness index [63] was used to evaluate the fairness of the algorithms.

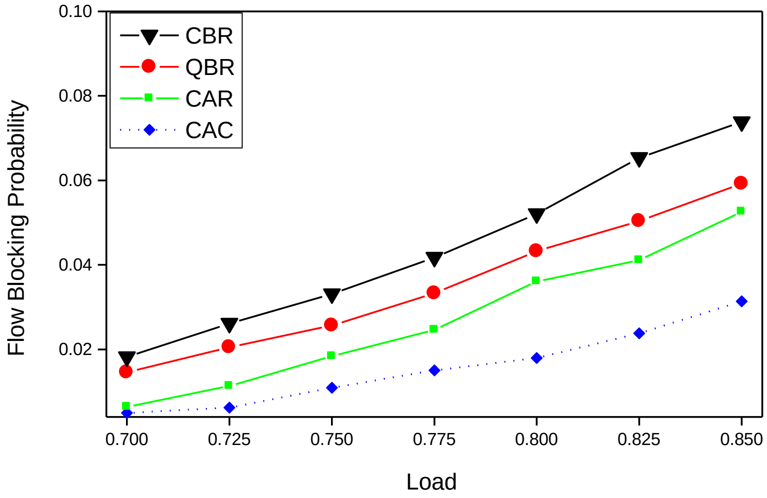
6.5.4 Simulation Results

6.5.4.1 Blocking Probabilities

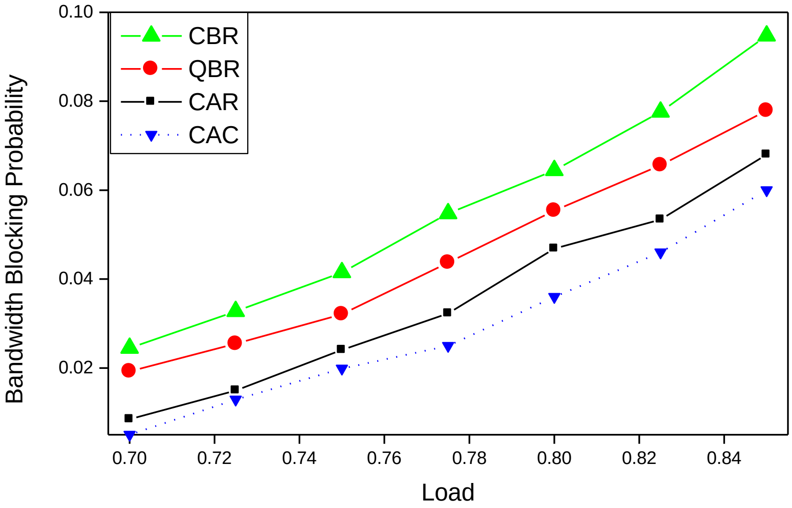
The performance of CBR, QBR, CAR and CAC algorithms are compared under different settings using flow blocking probability and bandwidth blocking probability.

6.5.4.2 Various Load Conditions

Figure 6.3 compares the performance of CBR and CAC algorithms in terms of flow blocking probability and bandwidth blocking probability which are plotted under various ranges of offered loads. The flow and bandwidth blocking probabilities are plotted against the offered load using the random topology Random 32.



(a) Flow Blocking Probability



(b) Bandwidth Blocking Probability

Figure 6.3 Performance of the CAC algorithm in Random 32 topology

We noted that under very low loads the difference in the performance of the four algorithms is relatively small, which is intuitively expected since the probability of finding sufficient resources in each link is high and flows are almost always accepted. However, performance varies significantly when the load increases. The bandwidth blocking probability grows more rapidly than the flow blocking probability, implying that flows with large bandwidth requirements are difficult to route, as expected.

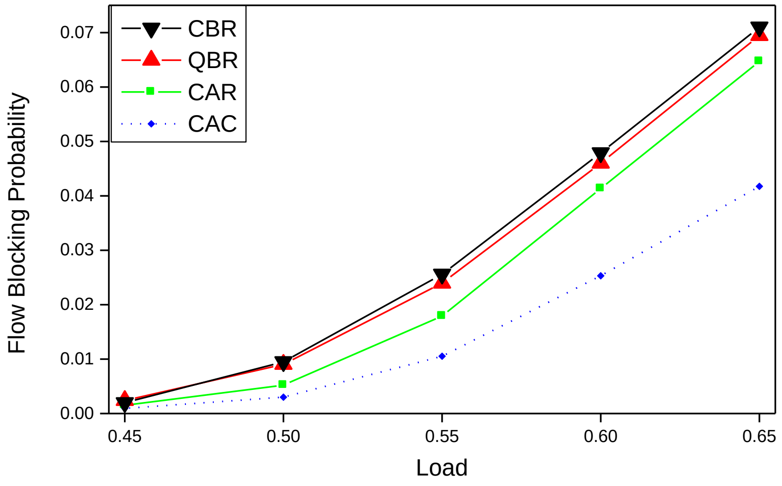
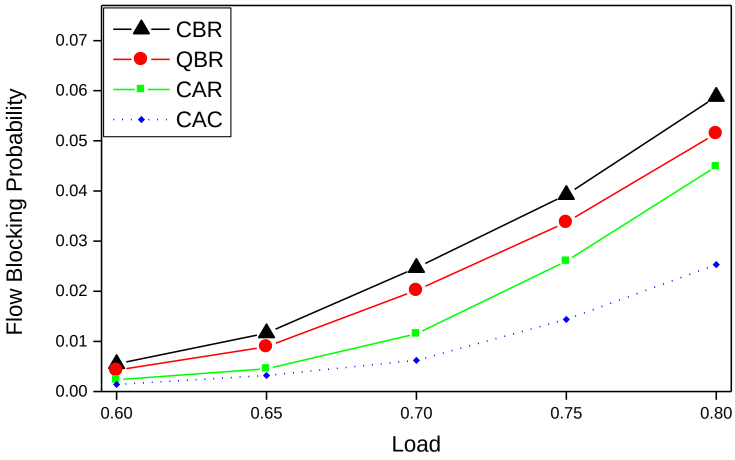
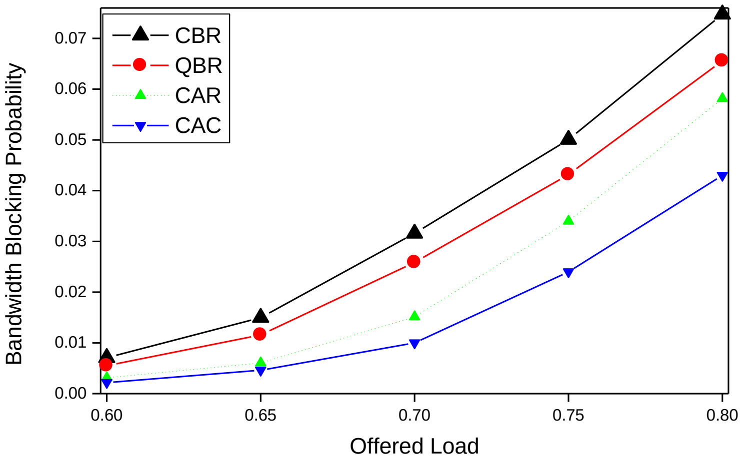


Figure 6.4 Performance of the CAC algorithm in ISP topology

It is also notable that the relative performance of the three algorithms is the same for flow and bandwidth blocking probabilities, suggesting that any may be used to evaluate the performance of Localised QoS routing algorithms, especially when the bandwidth requirements are slightly small compared to link capacities.



(a) Flow blocking probability



(b) Bandwidth blocking probability

Figure 6.5 Performance of CAC algorithm in scale-free topology

Since CAC makes routing decisions based on congestion information it gives the best performance compared with the other algorithms, it even performs better than CBR, particularly under heavy load.

The Localised CBR and QBR algorithms typically perform well in dense networks (Random 32 topology) in which there are enough candidate paths between any source and destination. On the other hand, Figure 6.4 and Figure 6.5 show the CAC algorithm in a sparse connected ISP topology and scale-free topology in which CBR and QBR show the worst performance compared with the CAC. The CAC algorithm gives superior performance and distributes the load throughout the network topology. The CAC algorithm clearly uses the more lightly loaded links and it is less dependent on the underlying network topology.

6.5.4.2 Load Balancing

Figure 6.6 shows the average Jain's index plotted as a function of the offered load under ISP topology. Each point in Figure 6.6 corresponds to one simulation run with the specified load. Since the index dynamically changes during the simulation with changes of instant carried traffic, the index is calculated at regular intervals of time over 2,000,000 connection requests. The load fluctuates rapidly over time and takes the average to be the Jain's index of the simulation run.

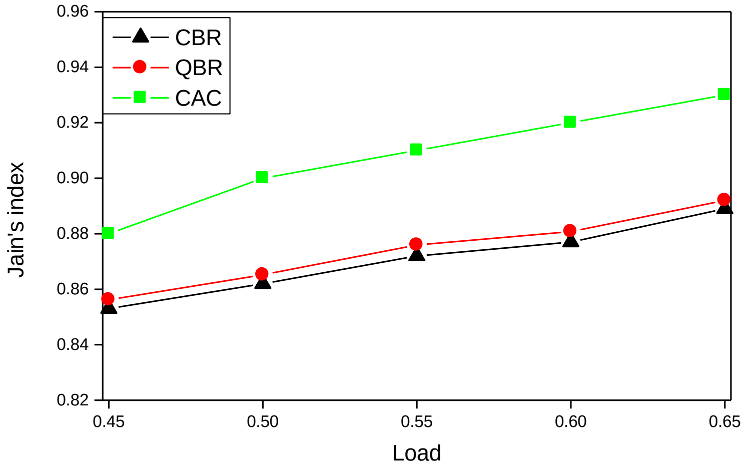


Figure 6.6 Average Jain's index in ISP topology

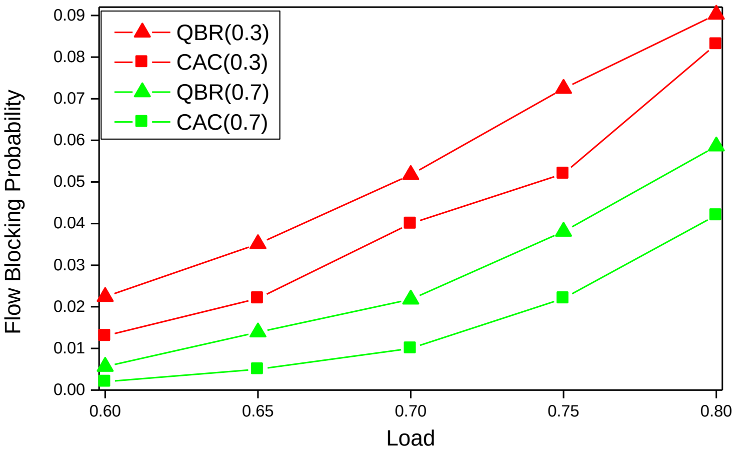
As shown in Figure 6.5, CAC gives the best fairness and always tries to distribute the incoming flows across all paths in the candidate set. It is shown that the fairness of the CAC algorithm increases as the load increases, as expected. Unlike CBR and QBR which always prefer short paths in sparse ISP topologies causing congestion, the CAC algorithm avoids congestion and distributes the load among the candidate paths; this can be noticed as it gives the best fairness index.

6.5.4.3 Impact of Bursty Traffic

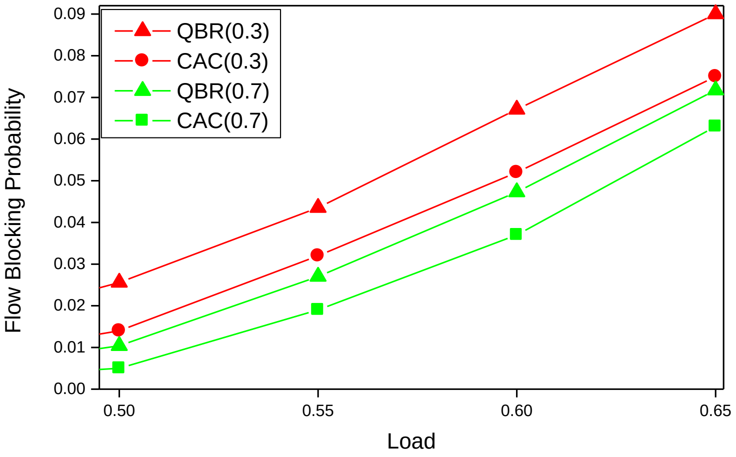
Although Poisson traffic models have been extensively used to characterize the various dynamics of the Internet flow arrivals, recent studies [94] [95] have showed that the flow arrivals with heavy tails, such as Weibull distributions, provide a more realistic traffic model representation. Consequently, the effect of bursty traffic is considered in the Scale-free topology and ISP topology where flow inter-arrival times follow the Weibull distribution.

Following [2] [94], the burstiness of the traffic is modelled with two different values of shape parameter for the distribution, namely, a=0.3 for large shape parameters and a=0.7 for small shape parameters where burstiness is increased with a smaller shape value.

Figure 6.7 shows the flow blocking probability plotted against the offered load with different shape values using scale-free topology and ISP topology. As shown in Figure 6.7, increased burstiness in the arrival process results in increased blocking probability over the offered loads used. Figure 6.7 shows that the burstiness has a significant impact on the performance of the CAC algorithm using different topologies and it outperforms the QBR in both shape parameters.



(a) Scale-Free Topology



(b) ISP Topology

Figure 6.7 Impact of bursty traffic

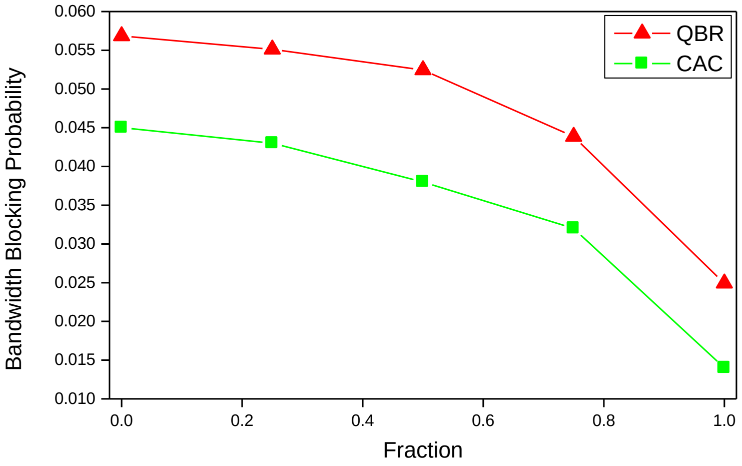
6.5.4.4 Large Bandwidth Requests

To examine the effect of large bandwidth requests, the case of a heterogeneous traffic is considered that contains two classes of flows, small-bandwidth flows and large-bandwidth flows but having the same flow duration. The amount of bandwidth requested by flows is derived randomly from the range [0.1-2MB], with mean $b_1=1.05$ for small-bandwidth flows, and [2-4] with mean $b_2=3$, for large-bandwidth flows. The flow duration for both bandwidth flows using scale-free topology is drawn from an exponential distribution with a mean value of 157 time units. The performance is measured by varying the mix of small- and large-bandwidth flows, *f* and *(1-f)* respectively; if the fraction=0 all the arriving flows are large-bandwidth flows, if the fraction=1 all the arriving flows are small-bandwidth flows, while keeping the offered load fixed. The arrival rates are changed according to the formula:

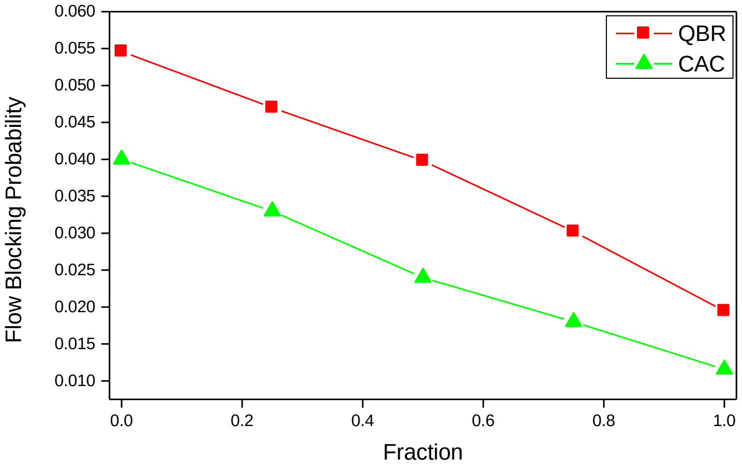
$$\rho = \frac{\lambda(f \times b_1 + (1-f) \times b_2)Nh}{\mu LC} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6.3)$$

Figure 6.8 shows the flow and bandwidth blocking probabilities plotted as a function of the fraction of small-bandwidth flows. As shown the QBR scheme gives the worst performance, regardless of the fraction of small-bandwidth flows. For both schemes, the performance degrades as the fraction of large-bandwidth flows increases. Consequently, as the fraction of small-bandwidth flows increases the blocking probability decreases and again, Figure 6.8 shows that the CAC algorithm performs better than QBR with small flows.

Figure 6.8 shows the difference between the flow blocking metric and the bandwidth blocking metric. The flow blocking probability decreases more rapidly than the bandwidth blocking probability, implying that flows with large bandwidth requirements are hard to route.



(a) Bandwidth Blocking Probability



(b) Flow Blocking Probability

Figure 6.8 Impact of large bandwidth in scale-free topology

The positive results presented in this chapter suggested that the use of call admission control with Localised routing using bandwidth metric is not necessarily scalable due to may increase the blocking probabilities at source nodes. Similarly, using call admission control using delay metric may a restrictive assumption due to the lack of scalability since the algorithm involves global information in each link / node [124].

However, following the researchers in the field of QoS routing algorithms, it is assumed to use a bandwidth as Quality of Service metric for this study due to the following reasons:

- Indeed, currently end-to-end delay is dominated by router hops and bandwidth rather than by delays in the routers [46] [86].
- Many QoS metrics can be expressed as a function of the bandwidth metric [83] [84] using rate based schedulers [80] [81] [82] as presented in the practical algorithms such as QOPSF [47].
- Using bandwidth avoids complexity and control efficiently link utilizations [85].

The use of call admission control which applied average / instantaneous delay accumulations for end-to-end delay need to be represented at the packet level.

However the objectives and procedures between flow level and packet level is different [125].

6.6 Time complexity

Most global QoS routing algorithms perform variants of Dijkstra's algorithm on the resulting graph. This takes at least O ($N \log N+L$) time, where N is the size of the network (nodes) and L is the number of links (edges) in the network. The time complexity of selecting a path among the set of candidate paths R in the CAC algorithm is O (|R|).

On the other hand, the CAC routing algorithm requires O (|1|).

In fact, QoS routing is scalable when the overhead increases linearly as the number of nodes increases. In global QoS routing each node requires a view on the status of the whole network through link state updates. On the other hand, in Localised CAC algorithm limits the decision at the edge only source nodes need to control the admission with routing. Moreover, Localised algorithms do not use the global state to maintain each router which imposes a routing overhead when keeping the link state updated.

6.7 Summary

In this chapter, different from previous Localised algorithms which select a path based on information updated from path request to path request, a Localised congestion avoidance routing algorithm (CAR) is presented which relies on making routing decisions at the source node, based on information updated in each connection request in order to avoid congestion and distributes the load efficiently. CAR algorithm made transition from static update information to dynamic update information. A simulation model was developed to enable a performance evaluation of the proposed algorithm in realistic scenarios.

A new call admission control with Localised QoS routing algorithm is developed in order to limit QoS routing decisions to edge routers and achieve an efficient utilisation of network resources. CAC algorithm reduces the overhead by minimizing the signalling effort [119] [122].

Only a portion of the results were observed in this chapter. The results were published in research papers [120] [121] [122], however many details had to be added for this chapter.

Chapter 7

Conclusions and Future Recommendations

7.1 Conclusions

One barrier to the deployment of large-scale networks is the scalability of QoS routing. The exchange of global state information has a critical impact on the path selection of QoS routing algorithms. However, it is impractical to maintain accurate network state information with rapid changes to network topology and growing network resources. The difficult elements of scalability are the overheads associated with maintaining the global state information and the overheads imposed by computation overheads. Any effective solution to the scaling problem should target at least one of these overheads.

A Localised QoS routing approach was proposed as an alternative approach to global QoS routing. This was proposed with the aim of solving the inherent problem of scalability in large-scale networks.

Such an efficient approach for Localised QoS routing has been the scope of this thesis. Throughout this thesis is demonstrated how Localised QoS routing algorithms could be considered an alternative scalable approach to the QoS routing problem. This thesis is concluded by highlighting the contributions of this thesis and revealing new insights into Localised QoS routing schemes. In the following section, further research is suggested or development that could be undertaken in the future.

- New Localised QoS routing algorithms are proposed. Each was compared with the Localised credit based QoS routing algorithm and it was shown that these could be considered viable alternative scalable approaches to the QoS routing problem.
- Throughout the thesis demonstrated that performance of any QoS routing algorithm depends on the network topology and the strictness of the QoS constraints. Hence, the network topology must be taken into account when evaluating the performance of any QoS routing algorithm.
- It is also demonstrated that the Localised QoS routing should be based on algorithms that directly reflect the quality of a path, rather than algorithms based indirectly on the quality of the path. A simulation model has been developed to enable the evaluation of the proposed performance in realistic scenarios.

- A new QBR algorithm is developed to make routing decisions using bandwidth, which is the same QoS metric used. It is demonstrated through simulations that they perform better than the CBR algorithm.
- A new BCM metric is developed to increase the flexibility of a large number of possible candidate paths that can be used to predict which network topologies gives better performance on the quality of Localised routing algorithms, taking path length into consideration.
- A new model (BCM network) is explored using the BCM metric that can be rewired to introduce increasing the connections that the network accepts.
- The results in this demonstrated that Localised QoS routing performs well with non-uniform traffic patterns, which lends itself to partitioned networks that reflect the non-uniformity using the BCM metric.
- A Localised congestion avoidance routing (CAR) algorithm is developed in which the decision of routing is taken based on information updated in each connection request which distributes the load efficiently. CAR algorithm made transition from static update information to dynamic update information.
- A new call admission control algorithm is developed in order to limit QoS routing decisions to edge routers and achieve an efficient utilisation of network resources.

- This thesis demonstrated that the presented Localised QoS routing scheme performs better than the CBR algorithm with bursty connection requests, heterogeneous traffic and, therefore, it might be easy to deploy in practice on realistic networks.

7.2 Recommendations for Future Research

Some logical recommendations listed for future research work that could be considered within the context of this thesis.

- The proposed Localised routing algorithms assumed the use of bandwidth as a QoS metric and that the performance of Localised QoS routing algorithms depends on the selection of candidate path set; this needs more investigation in other metrics. Another feasible extension to the proposed algorithms would combine bandwidth and delay constraints in future research work.
- Extending the Localised algorithms to the case of QoS multicast routing is another important area. No multicast algorithm has been developed based on this Localised routing approach.
- The issue of selecting a candidate path set has a significant impact on how Localised QoS routing algorithms perform, and this needs to be investigated

further in the context of the dynamic selection of a candidate path set between any source and destination in the topology.

- A call admission control approach is needed with regard to Localised routing algorithms to route guaranteed traffic for multimedia applications.
- Considering and generating the BCM topology by distribution probability function analysis is a possible future work.
- Realistic topologies such as the Internet are not static networks. Evaluation of the BCM clustering metric by increasing the number of nodes is also required from future research work.

References

[1] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell and J. McManus, “Requirements for Traffic Engineering over MPLS,” *RFC-2702,* September 1999.

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

[3] A. S. Tanenbaum, *Computer networks,* 4th ed. Upper Saddle River, N.J.: Prentice Hall PTR, 2003.

[4] D. P. Bertsekas and R. G. Gallager, *Data networks,* 2nd ed. Englewood Cliffs, N.J.: Prentice Hall, 1992.

[5] J. Moy, “OSPF Version 2,” in *Internet Request for Comments (RFC 2328),* 1998.

[6] G. Malkin, “RIP version 2,” in *IETF RFC 2453,* 1998.

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

[8] J. Postel, “Transmission control protocol,” in *IETF RFC 793,* 1981.

[9] J. Postel, “User datagram protocol,” in *IETF RFC 768,* 1980.

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

[11] X. Masip-Bruin et al., “Research Challenges in QoS Routing,” *Computer Communications,* vol. 29, pp. 563-581, 2006.

[12] R. Braden, D. Clark, and S. Shenker, “Integrated Services in the Internet Architecture: An Overview,” in *IETF RFC 1633,* 1994.

[13] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, “Resource ReSerVation Protocol (RVSP) – Version 1 Functional Specification,” in *IETF RFC 2205,* 1997.

[14] S. Blake et al. “An Architecture for Differentiated Services,” in *IETF RFC 2475*, 1998.

[15] E. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol Label Switching Architecture,” in *Internet Draft<draft-ietf-mpls-arch-01.txt>*, 1998.

[16] D. Durham and R. Yavatkar, *Inside the Internet’s Resource ReSerVation Protocol: Foundations for Quality of Service*, John Wiley & Sons, 1999.

[17] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*: Morgan Kaufmann, 2001.

[18] A. Viswanathan, N. Feldman, Z. Wang, and R. Callon, “Evolution of multiprotocol label switching,” *IEEE Communications Magazine*, vol. 36, no. 5, pp. 165-173, May 1998.

[19] O. Younis and S. Fahmy, “Constraint-Based Routing in the Internet: Basic Principles and Recent Research,” *IEEE Communications Surveys & Tutorials*, vol. 5, pp. 2-13, 2003.

[20] S. Chen and K. Nahrstedt, “An Overview of Quality-of-Service Routing for the Next Generation High Speed Networks: Problems and Solutions,” *IEEE Network Magazine*, vol. 12, pp. 64 -79, 1998.

[21] S. Srivastava et al. “Benefits of Traffic Engineering using QoS Routing Schemes and Network Controls,” *Computer Communications Journal*, vol. 27, pp. 387-399, 2004.

[22] H. Lu and I. Faynberg, “An Architectural Framework for Support of Quality of Service in Packet Networks,” *IEEE Communications Magazine*, pp. 98-105, 2003.

[23] Z. Whang and J. Crowcroft, “Quality-of-Service routing for supporting multimedia applications,” *IEEE J. Select. Areas Communication*, vol. 14, pp. 1228-1234, 1996.

[24] S. Nelakuditi, R. Tsang, and Z. Zhang, “Quality-of-Service Routing without Global Information Exchange,” in *IWQOS*, 1999.

[25] ATM Forum, “Private Network-Network Interface Specification Version 1.1 (PNNI 1.1),” April 2002.

[26] L. Kleinrock and F. Kamoun, *Hierarchical Routing for Large Networks: Performance Evaluation and Optimization*, Computer Networks, vol. 1, No. 3, pp. 155-174, January 1977.

[27] R. Guerin and A. Orda, "QoS-Based Routing in Networks with Inaccurate Information: Theory and Algorithms", *IEEE INFOCOM*, Japan, April 1997.

[28] N. Ansari, G. Cheng, and N. Wang, "Routing-oriented update scheme (ROSE) for link state updating," *IEEE Transactions on Communications*, vol. 56, pp. 948-956, 2008.

[29] B. Fu, F. A. Kuipers, and P. V. Mieghem, "To update network state or not?" in *The 4th international telecommunication networking workshop on QoS in multiservice IP networks*, Feb 2008.

[30] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, pp. 1: 269-271, 1959.

[31] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1957.

[32] R. M. Fano, "A Heuristic Discussion of Probabilistic Decoding," *IEEE Transactions on Information Theory*, vol. 64, no. 9, April 1963.

[33] N. Seitz, "ITU-T QoS Standards for IP-Based Networks," *IEEE Communications Magazine*, pp. 82-89, June 2003.

[34] C-Q. Yang and Alapati V. S. Reddy, "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks," *IEEE Network*, pp. 34-45, July/August 1995.

[35] Q. Ma and P. Steenkiste, "Quality-of-Service Routing for Traffic with Performance Guarantees," in *Proceedings of the IFIP Fifth International Workshop on Quality of Service*, New York, pp. 115-126, 1997.

[36] R. Guerin and A. Orda, "QoS based Routing in Networks with Inaccurate Information: Theory and Algorithms," *IEEE/ACM Transactions on Networking*, 7, pp. 605-614, 1999.

[37] J. L. Sobrinho, "Algebra and Algorithms for QoS Path Computation and Hop-by-Hop Routing in the Internet," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 541-550, 2002.

[38] D. S. Reeves and H. F. Salama, "A Distributed Algorithm for Delay-Constrained Unicast Routing," *IEEE/ACM Transactions on Networking,* April 2000.

[39] B. Awerbuch, B. K. Y. Du, and Y. Shavitt, "Routing Through Teranode Networks with Topology Aggregation," in *IEEE ISCC'98*, Athens, Greece, 1998.

[40] J. Bhhrens and J. J. Garcia-Luna-Aceves, "Hierarchical Routing Using Link Vectors," in *IEEE Infocom'98*, 1998.

[41] R. Izmailov, A. Iwata, B. Sengupta, and H. Suzuki, "PNNI Routing Algorithms for Multimedia ATM Internet," *NEC Research and Development*, vol. 38, pp. 60-73, 1997.

[42] A. Iwata and N. Fujita, "A Hierarchical Multilayer QoS Routing System with Dynamic SLA Management," *IEEE Journal on Selected Areas in Communication,* vol. 18, pp. 2603–2616, December 2000.

[43] T. Korkmaz and M. Krunz, "Source-oriented topology aggregation with multiple QoS parameters in hierarchical networks," *ACM Transactions on Modeling and Computer Simulation,* vol. 10, pp. 295--325, 2000.

[44] M. Curado and E. Monteiro, "A Survey of QoS Routing Algorithms," in *the International Conference on Information Technology (ICIT2004)*, Istanbul, Turkey, December 2004.

[45] A. Gibbons, *Algorithmic Graph Theory*. Cambridge: Cambridge University Press, 1985.

[46] B. Peng, A. H. Kemp, and S. Boussakta, "QoS Routing with bandwidth and Hop-Count Consideration: A Performance Perspective," *Journal of Communications,* vol. 1, 2006.

[47] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda, and T. Przygienda., "QoS Routing Mechanism and OSPF Extensions," in *RFC 2676*, 1999.

[48] A. A. Shaikh, "Efficient Dynamic Routing In Wide-Area Networks" in *Dept. of Computer Science and Engineering,* Ann Arbor: The University of Michigan, 1999.

[49] I. Matta and A. U. Shankar, "Dynamic routing of real-time virtual circuits," in *Proceedings of IEEE International Conference on Network Protocols*, Columbus, OH, pp. 132-139, 1996.

[50] A. Shaikh, J. Rexford, and K. Shin, "Evaluating the Overheads of Source-Directed Quality of Service Routing," in *Proceedings of the International Conference on Network Protocols (ICNP)*, 1998.

[51] A. Shaikh, J. Rexford, and K. Shin, "Efficient Pre-computation of Quality of Service Routes," in *International Workshop on Network and OS Support for Digital Audio and Video (NOSSDAV '98)* Cambridge, 1998.

[52] Q. Ma and P. Steenkiste, "On path selection for traffic with bandwidth guarantees," in *5th IEEE International Conference on Network Protocols*, Atlanta, GA, pp. 191-202, 1997.

[53] Q. Ma, P. Steenkiste, and H. Zhang, "Routing High-bandwidth Traffic in Max-min Fair Share Networks," in *Proceedings of ACM SIGCOMM'96,* Palo Alto, CA, pp. 206-217, 1996.

[54] S. Nelakuditi, "Localised approach to providing quality-of-service," in *Dept. of Computer Science and Engineering* Minneapolis: University of Minnesota, p. 174, 2001.

[55] S. Nelakuditi, Z. L. Zhang, R. Tsang, and D. Du, "Adaptive Proportional Routing: a Localised QoS Routing Approach," *IEEE/ACM Transactions on Networking,* vol. 10, pp. 790-804, 2002.

[56] X. Yuan and A. Saifee, "Path Selection Methods for Localised Quality of Service Routing," in *The 10th IEEE International Conference on Computer Communications and Networks (IC3N 2001),* Phoenix, Arizona, pp. 102-107, 2001.

[57] S. Nelakuditi, Z.-L. Zhang, R. Tsang, and D. Du, "On selection of candidate paths for proportional routing," *Computer Networks,* vol. 44, pp. 79-102, 2004.

[58] S. Alabbad and M. E. Woodward, "Localised Credit Based QoS Routing: Performance Evaluation Using Simulation," in *the 40th Annual Simulation Symposium*, Huntsville, AL. 2006.

[59] P. Paul and S. V. Raghavan, "Survey of QoS routing," in *15th international conference on Computer communication*, Mumbai, India, pp. 50-75, 2002.

[60] A. Varga, "OMNeT++ Community Site," in *URL:http://www.omnetpp.org*, 2010.

[61] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *the European Simulation Multi-conference*, Prague, Czech Republic, 2001.

[62] A. Varga, "OMNeT++ Discrete Event Simulation System Version 3.2 User Manual," in *URL:http://www.omnetpp.org/doc/manual/usman.html*, 2008.

[63] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination For Resource Allocation In Shared Computer Systems," *DEC Research Report TR-301*, 1984.

[64] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 3rd ed. New York: McGraw-Hill, 2000.

[65] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an inter-network," in *Proceedings of IEEE INFOCOM*, pp 494-602, March 1996.

[66] V. Paxson and S. Floyd, "Why we don't know how to simulate the Internet," in *Proceedings of the 1997 Winter Simulation Conference*, 1997.

[67] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Transactions on Networking*, 9, pp 392-403, 2001.

[68] H. Pung, J. Song, and L. Jacob, "Fast and efficient flooding based QoS routing algorithm," in *IEEE ICCCN99*, pp. 298-303, 1999.

[69] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service routing in High Speed Networks Based on Selective Probing," *23rd Annual Conference on Local Area Networks (LCN'98)*, 1998.

[70] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An Approach to Universal Topology Generation," *The International Workshop on Modeling Analysis and Simulation of Computer and Telecommunications Systems*, Cincinnati, Ohio, 2001.

[71] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Select. Areas Communications*, vol. 6(9), pp. 1617-1622, 1988.

[72] B. Peng, A. Kemp, and S. Boussakta, “Impact of network conditions on QoS routing algorithms,” in *3rd IEEE Consumer Communications and Networking*, pp. 25-29, 2006.

[73] K. Kowalik and M. Collier, “Should QoS routing algorithms prefer shortest paths?”, *IEEE International Conference on Communications*, Anchorage Alaska, May 2003.

[74] J. Moy, “OSPF Version 2,” in *Internet Request for Comments (RFC 2328)*, 1998.

[75] J. T. Moy, *OSPF: Anatomy of an Internet routing protocol*. Reading, Mass.: Addison-Wesley, 1998.

[76] J. T. Moy, *OSPF complete implementation*. Boston: Addison-Wesley, 2001.

[77] *Private Network-Network Interface (PNNI) v1.0 specifications*, ATM Forum, 1996.

[78] S. Alabbad and M. E. Woodward, “Localised Credit Based QoS Routing,” *Proceedings of IEE,* vol. 153, pp. 787-796, 2006.

[79] R. J. Gibbens, F. P. Kelly, and P. B. Key, “Dynamic Alternative Routing: Modelling and Behaviour,” in *12th Int. Teletraffic Congress*, Torino, 1988, pp. 3.4A.3.1--3.4A.3.7.

[80] A. Demers, S. Keshav, v. S. Shenker, “Analysis and Simulation of a Fair Queuing Algorithm,” *Journal of Internetworking Research and Experience*, vol. 1, pp. 3-26, 1990.

[81] S. Rampal and D. Reeves, “An Evaluation of Routing and Admission Control Algorithms for Multimedia Traffic,” *Computer Communications Journal*, vol. 18, pp. 332-345, 1997.

[82] L. Georgiadis, R. Guérin, V. Peris, and K. N. Sivarajan, “Efficient network QoS provisioning based on per node traffic shaping,” *IEEE/ACM Transactions on Networking* vol. 4, pp. 482-501, 1996.

[83] S. Shenker, C. Partridge, and R. Guerin, “Specification of guaranteed quality of service,” in *RFC no. 2212, Internet RFC*, 1997.

[84] H. Zhang, “Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks”, *Proceedings of the IEEE,* 83(10), Oct. 1995.

[85] X. Xiao, et al. “Internet QoS: A Big Picture”, *IEEE Network*, Mar/April 1999.

[86] J. F. Kurose and K. W. Ross, *A top-down approach featuring the Internet* Boston; London: Pearson/ Addison Wesley, 2005.

[87] D. H. Lorenz and A. Orda, “Optimal Partition of QoS Requirements on Unicast Paths and Multicast Trees,” *IEEE/ACM Transactions on Networking*, vol. 10, pp. 102-114, 2002.

[88] S. Chen and K. Nahrstedt, “Distributed Quality-of-Service routing in High Speed Networks Based on Selective Probing,” in *23rd Annual Conference on Local Area Networks (LCN'98)*, 1998, pp. 80--89.

[89] G. Apostolopoulos, R. Guerin, S. Kamat, S. K. Tripathi, “Improving QoS Routing Performance Under Inaccurate Link State Information”, *Proc. ITC’16*, June 1999.

[90] X. Masip-Bruin, E. Marin-Tordera, M. Yannuzzi, R. Serral-Garcia and S. Sanchez-Lopez, “Reducing the effects of routing inaccuracy by means of prediction and an innovative link-state cost” *IEEE Communications letters*, vol. 14, no. 5, May 2010.

[91] S. Kim and M. Lee, “Server based QoS routing with implicit network state updates,” in *Proceedings of IEEE* Globecom, San Antonio, TX, November 2001.

[92] T. Anjali, “A new path selection algorithm for MPLS networks based on available bandwidth estimation,” in *Proceedings of QofIS 2002*, October 2002.

[93] N. Spring, R. Mahajan and D. Wetherall, “Measuring ISP Topologies with Rocketfuel”, *Proceedings of ACM/SIGCOMM’02*, August 2002.

[94] A. Feldmann, “Impact of non-poisson arrival sequences for call admission algorithms with and without delay,” in *GLOBECOM'96. IEEE Global Telecommunications Conference*, London, UK, pp. 617-622, 1996.

[95] A. Feldmann, “Characteristics of TCP connections arrivals,” in *Self-similar Network Traffic and Performance Evaluation*, K. Park and W. Willinger, Eds. New York: John Wiley and Sons, pp. 367-399, 2000.

[96] S. Alabbad and M. E. Woodward, “Localised Route Selection for Traffic with Bandwidth Guarantees,” *SIMULATION,* vol. 83, pp. 259-272, 2007.

[97] R. Gawlick, A. Kamath, S. Plotkin, and K. Ramakrishnan, “Routing and Admission Control in General Topology Networks,” *Technical Report STAN-CS-TR-95-1548,* 1995.

[98] K. Kowalik, M. Collier, “Connectivity Aware Routing - a Method for Finding Bandwidth Constrained Paths Over a Variety of Network Topologies”. *IEEE International Symposium on Computers and Communications,* pp. 392-398 July 2003.

[99] S. H. Strogatz, “Exploiting complex networks,” *Nature,* vol. 410, pp 268-276, March 2001.

[100] R. Albert and A-L. Barabasi, “Statistical mechanics of complex networks,” *Review of Modern Physics,* vol. 74, pp.47-91, January 2002.

[101] Z. Wang, A. Scaglione, R. Thomas, “Generating Statistically Correct Random Topologies for Testing Smart Grid Communication and Control Networks”. *IEEE Transactions on Smart Grid*, vol. 1, no. 1, June 2010.

[102] P. Erdős and R. Rényi, “The evolution of Random Graphs”, *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17-61, 1960.

[103] D. J. Watts, S. H. Strogatz, “Collective Dynamic of Small World Networks,” *Nature,* 393 (6684): pp. 409-10, 1998.

[104] A-L. Barabási, R. Albert, “Emergence of Scaling in Random Networks,” *Science,* 286: pp. 509-512, 1999.

[105] X. F. Wang and G. Chen, “Complex Networks: Small-World, Scale-Free and Beyond,” *IEEE Circuits and Systems Magazine*, pp. 6-20, 1st Quarter, 2003.

[106] C. Faloutsos, P. Faloutsos and M. Faloutsos, “On Power-Law Relationship of the Internet Topology,” *Proceedings of the ACM SIGCOMM,* September 1999.

[107] S. Kim and K. Harfoush, “The Causal Factors Behind Internet Power-Law Connectivity”, in *Proceedings of IEEE Conference of Computer Communications INFOCOM'07,* Anchorage, Alaska, May 2007.

[108] S. Boccaletti and V. Latora “Complex networks: Structure and Dynamics”, *Physical Reports*, 424: pp. 175-308, 2006.

[109] Y. Li, J. H. Cui, D. Maggiorini and M. Faloutsos, “Characterizing and Modelling Clustering Features in AS level Internet Topology” *The 27th Conference on Computer Communications. IEEE INFOCOM'08*, pp. 271-275, April 2008.

[110] T. Bu and D. Towsley, “On distinguishing between Internet power law topology generators,” in *Proceedings of IEEE Conference of Computer Communications INFOCOM'02*, New York, June 2002.

[111] P. Fraigniaud, “A New Perspective on the Small-World Phenomenon: Greedy Routing in Tree-decomposed Graphs,” in *Proceedings of 13th Annual European Symposium on Algorithms (ESA)*, pp. 791-802, 2005.

[112] P. Mahadevan, D. Krioukov, K. Fall and A. Vahdat. “Systematic Topology Analysis and Generation using Degree Correlations,” in *Proceedings of ACM SIGCOMM*, pp. 135-146, 2006.

[113] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks,* Oxford University Press 2003.

[114] J. Soldatos, E. Vayias, and G. Kormentzas, “On the building blocks of quality of service in heterogeneous IP networks,” *IEEE Communications Surveys and Tutorials,* vol. 7, 2005.

[115] A. Vazquez, R. Pastor-Satorras and A. Vespignani, *Internet Topology at the Router and Autonomous System Level,* arXiv: cond-mat/0206048 v1, June 2002.

[116] S. Nelakuditi and Z-L Zhang, *Localised Quality of Service Routing for the Internet,* Kluwer Academic Publishers, 2003.

[117] 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.

[118] A. H. Mohammad and M. E. Woodward, “Localised Quality Based QoS Routing,” *Proceedings of International Symposium on Performance of Computer and Communications Systems*, IEEE Press, SPECTS, June 2008.

[119] A. H. Mohammad and M. E. Woodward, “Effects of Topology on the Performance of Localised QoS Routing Algorithms,” *Proceedings of International Symposium on Performance of Computer and Communications Systems*, IEEE Press, SPECTS, July 2009.

[120] A. H. Mohammad and M. E. Woodward, “Localised QoS Routing and Admission Control: Performance Evaluation,” *Proceedings of 6th International Conference on Innovations on Information Technology*, IEEE Press, December 2009.

[121] A. H. Mohammad and M. E. Woodward, “Localised QoS Routing and Admission Control for Congestion Avoidance,” *Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems*, IEEE Press, CISIS, February 2010.

[122] A. H. Mohammad. “New Localised Call Admission Control Algorithms in Communication Networks with Quality of Service Constraints”, International Journal of Computer Science and Network Security, Vol.10, No.11, pp. 125-131, November 2010.

[123] Al Ghamdi T. A. Woodward, M. E. "Novel algorithms for QoS Localised routing in communication networks", First Asian Himalayas International Conference on Internet AH-ICI 2009.

[124] Alzahrani A. "Localised Quality of Service Routing Algorithms for Communication Networks" Ph D thesis University of Bradford 2009.

[125] A. H. Mohammad. “A new Localised Network based Routing Model in Computer and Communication Networks”, International Journal of Computer Networks & Communications, Vol.3, No.2 pp. 146-163, March 2011.