Evaluation of Performance of QoS (Quality of Service) Routing Strategy for the Differentiated Services Framework by Simulation

Dissertation submitted to *Jawaharlal Nehru University* in the partial fulfillment of the requirements for the award of Degree of

Master of Technology
In
Computer Science & Technology
By
KULWANT RAI

SCHOOL OF COMPUTER & SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-67

JANUARY 2004

Evaluation of Performance of QoS (Quality of Service) Routing Strategy for the Differentiated Services Framework by Simulation

Dissertation submitted to ***Jawaharlal Nehru University*** in the partial fulfillment of the requirements for the award of Degree of

Master of Technology
In
Computer Science & Technology
By
KULWANT RAI

SCHOOL OF COMPUTER & SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-67

JANUARY 2004

CERTIFICATE

This is to certify that the dissertation entitled **"Evaluation of Performance of QoS (Quality of Service) Routing Strategy for the Differentiated Services Framework by Simulation"** which is being submitted by **Mr. Kulwant Rai** to the School of Computer & Systems Sciences, Jawaharlal Nehru University New Delhi, in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science & Technology**, is a record of bona-fide work carried out by him under the guidance and supervision of Prof. G. V. Singh, during the Monsoon Semester, 2003.

The matter embodied in this dissertation has not been submitted in part or full to any other university or institution for the award of any degree.

Prof. Karmeshu
Dean, SC&SS,
JNU, New Delhi-67

Prof. G. V. Singh
(Supervisor)
05/01/2004

DECLERATION

This is to certify that the dissertation entitled "**Evaluation of Performance of QoS (Quality of Service) Routing Strategy for the Differentiated Services Framework by Simulation**" which is being submitted to School of Computer & Systems Sciences, Jawaharlal Nehru University, for the award of degree of **Master of Technology in Computer Science & Technology**, is a record of bonafide work carried out by me.

This work is original and has not been submitted in part or full to any university or institution for the award of any degree.

[KULWANT RAI]

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people whose ceaseless cooperation made it possible and whose guidance and encouragement crown all efforts with success.

I would like to sincerely thank my supervisor **Prof. G.V. Singh**, School of Computer and Systems Sciences, Jawaharlal Nehru University for his able guidance, encouragement and support extended by him in successful completion of this project work. Their ideas were innovative and the valuable discussions we had, was very helpful in keeping the project on right track.

I would like to record my sincere thanks to dean, **Prof. Karmeshu** for providing necessary facilities.

I would also like to thank **Mr. Jugal Kishore** for the support, guidance and valuable ideas, which he provided throughout the project.

Last but not the least I would like to thank all faculty members for their encouragement and guidance.

[KULWANT RAI]

…dedicated to my family

ABSTRACT

Quality of Service aims to improve the satisfaction of user by treating traffic with special requirements differently as well as improving the utilization of resources. QoS routing computes paths that are subject to QoS requirements and can improve the utilization of the network resources. Besides benefits, QoS routing also results in cost. The cost of QoS routing includes path computation cost, link state update cost and storage cost.

In the present work, we have evaluated performance of QoS routing relative to communication overhead, which in turn affects processing overhead. We evaluated the efficiency of thresholds and moving average, which are mechanisms to reduce communication overhead induced by the dynamic nature of QoS routing. The impact of different network topologies and traffic patterns is also studied.

Our results show that these mechanisms are effective in controlling the communication overhead and network topology and traffic load have direct impact on communication overhead. A simulator using C++ language was written to achieve these objectives. The simulator provides facilities to specify various parameters using input files and output files generated are analyzed by drawing various graphs using EXCEL.

CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Background

With the rapid transformation of Internet into commercial infrastructure, demands for service quality have rapidly developed. Internet is expected to support not only traditional services (e.g. email, ftp etc) but also the upcoming high speed and real time services (e.g. audio/video, real time transmission and virtual private networks etc). The later services represent much different traffic characteristics from the former services in terms of bit rate, delay and loss they can tolerate and they require fixed assurance of Quality of Service (QoS) in duration of transmission.

These specific requirements call for mechanisms that can provide differential treatment to more demanding applications along with best effort service to those requiring only connectivity. These mechanisms, called quality of service (QoS) mechanisms aims at managing resources more effectively to meet wide range of application requirements.

However, the current Internet does not support QoS requirements. As a result, need for a high performance network emerges. It is almost impossible to build a new high performance network while abandoning the legacy networks e.g. public switched telephone network (PSTN) and Internet. The sole solution seems to be converting the existing networks into future networks under a general architecture. Two strategies for achieving the goal are proposed. The first one is that fibers and wavelength division multiplexing (WDM) will make bandwidth so abundant and cheap that QoS will be automatically delivered. The other one is to build up a QoS-based network on the basis of the current Internet by providing classified services with quality requirements guaranteed. Even if the bandwidth will eventually become abundant and cheap, it is not going to happen soon. For now, some simple mechanisms are definitely needed in order to provide QoS on the Internet.

It is apparent that several service types will be demanded from the Internet in future. One service type will provide predictable Internet services for companies that do business on the Web. Such companies will be willing to pay a certain price to make their services reliable and give their users a fast feel of their websites [15]. Another type of service will be to provide low delay and low jitter services to the applications such as Internet Telephony and videoconferencing. Meanwhile, best effort service will remain for those customers who need only connectivity. Therefore the current Internet is expected to become a QoS based Internet in which various services with QoS requirements will be provided.

1.2 Objectives of Quality of Service

QoS refers to the ability of a network to provide better services to selected traffic over different underlying technologies. QoS features provide better and more predictable network services by:

- Supporting dedicated bandwidth
- Improving loss characteristics
- Avoiding and managing network congestion
- Shaping network traffic
- Setting traffic priorities across the network

The objective of QoS architecture is to provide a framework for the integration of quality of service control and management mechanisms. The following three components are necessary to deliver QoS across a network:

- QoS within a single network element, which includes queuing, scheduling and traffic shaping etc.
- QoS signaling techniques for coordinating QoS from end-to-end between network elements.
- QoS policing and management function to control and administer end-to-end traffic across a network.

To configure QoS features throughout a network and to provide quality service delivery, a general QoS architecture as well as some mechanisms such as traffic engineering are needed. So far many different kinds of QoS architectures and mechanisms have been proposed to meet the demand of QoS. Quality of service that is required for traffic may be specified by the following parameters:

- Service availability
- End-to-end Delay
- Delay Variation
- Throughput
- Packet loss

Different applications vary in the required values of these parameters for their acceptable performance.

1.3 QoS in the Internet

Present day Internet provides “Best Effort Service”. Under best effort scheme, the Internet treats all packets equally. Traffic is processed as quickly as possible but there is no quantified guarantee as to timeliness or actual delivery. In the times of congestion packets are dropped randomly without regard to relative importance or timeliness requirements of traffic. As the volume of traffic grows all packet delivery slows down and the service quality degrades, but service is never denied.

According to requirements of traffic, Internet traffic can be classified [1] into two broad categories: Elastic traffic and Inelastic traffic. These types significantly differ in their requirements. Elastic traffic can adjust, over wide ranges, to changes in delay and throughput across an internet and still meet the needs of applications. This is type of traffic for which internets were designed. Elastic traffic includes common Internet applications such as electronic mail, file transfer, remote logon and web access etc. Inelastic traffic do not easily adapt to changes in delay and throughput across an internet. Examples of inelastic traffic are real-time traffic such as voice and video. The requirements for inelastic traffic may include demand for minimum throughput, maximum bound for packet loss or delay variation (jitter) that can be tolerated.

These requirements are difficult to meet in an environment with variable queuing delays and congestion losses. Inelastic traffic introduces new requirements into internet architecture. Some means are needed to give preferential treatment to applications with more demanding applications. Quality of service aims to provide some level of predictability and control beyond current IP's "Best Effort Service".

Many efforts have been put forward to provide guaranties for the specific services or customers, e.g. Integrated Services (IntServ) with the Resource Reservation Protocol (RSVP) and the Differentiated Services (DiffServ) architecture, Multi-Protocol Label Switching (MPLS), quality of service routing and traffic engineering etc. MPLS is a forwarding scheme. It can be used together with DiffServ to provide better QoS. Quality of Service routing is to compute paths that are subject to QoS requirements. On the other hand, traffic engineering is concerned with performance optimization of operational networks.

1.3.1 Integrated Services

The fundamental idea of IntServ architecture is to reserve resources such as bandwidth and buffers, for a given flow to ensure that the QoS required by the flow is satisfied. In addition to best effort service, IntServ offers two other services: guaranteed service and controlled load service.

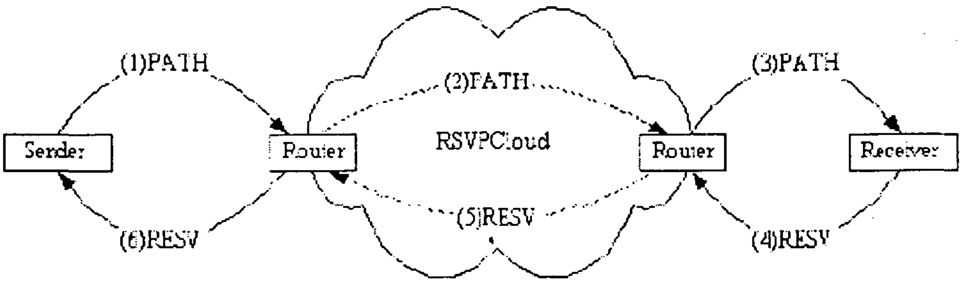


Fig 1. RSVP Signaling

RSVP was invented as a signaling protocol for applications to reserve resources. The signaling process is illustrated in Fig1. The sender sends a PATH message to the receiver specifying the characteristics of the traffic. Every intermediate router along the path forwards the PATH message to the next hop determined by the routing protocol. Upon receiving a PATH message, the receiver responds with a RESV message to request resources for the flow. Every intermediate router along the path can reject or accept the request of the RESV message. If the request is rejected, the router will send an error message to the receiver, and the signaling process will terminate. If the request is accepted, link bandwidth and buffer space are allocated for the flow and the related flow state information will be installed in the router. Integrated Services is implemented by following components:

1.3.1.1Admission control

Admission control implements the decision algorithm that a router or host uses to determine whether a new flow can be granted without impacting earlier guarantees. Admission control is invoked at each node to make a local accept/reject decision when a host requests a real time service along some path through the Internet. In addition to ensuring that QoS guarantees are met, admission control is concerned with enforcing administrative policies on resource reservations. Some policies will demand authentication of those requesting reservations. Besides, admission control plays an important role in accounting and administrative reporting.

1.3.1.2 Packet Classifier

Packet classifier maps each incoming packet into some classes. Choice of classification may be based on the contents of the existing packet headers or some additional classification number added to each packet.

1.3.1.3 Packet Scheduler

The packet scheduler manages the forwarding of different packet streams using a set of queues and other mechanisms. All packets in the same class get the same treatment from packet scheduler.

1.3.1.4 Service classes

Guaranteed Service guarantees that data-grams will arrive within the guaranteed delivery time and will not be discarded due to queue overflows, and ensures that flow's traffic stays within specified parameters. This service is intended for applications that need firm guarantee that a data-gram will arrive no later than a certain time after it was transmitted by source. Real rime applications like audio/video applications need such services.

Controlled Load Service intends to support a broad class of applications that were originally developed for Internet, but highly sensitive to network overload. Controlled load service strives to approximate tightly the behavior visible to applications receiving best effort service during unloaded conditions. This implies that both packet loss ratio and minimum delay will remain unchanged regardless of the overall load level within the network

To ensure that these requirements are fulfilled, subscribers requesting controlled load service initially provide the intermediate network elements with an estimation of the data traffic that they will generate. If the traffic generated by the subscriber falls outside of the region described by this description the QoS provided to the subscriber may deteriorate in terms of delay and packet loss.

The Integrated Services/RSVP architecture represents fundamental change to the current Internet architecture, which is founded on the concept that all flow related state information should be in the end systems. Some problems with the Integrated Services Architecture are:

- The amount of state information increases proportionally with the number of flows. This places a huge storage and processing overhead on routers. Therefore, this architecture is not scalable.
- The requirement on routers is high. All routers must implement RSVP, admission control, MF classification and packet scheduling.
- Ubiquitous deployment is required for Guaranteed Service. Incremental deployment of Controlled-Load Service is possible by deploying Controlled-Load Service and RSVP functionality at the bottleneck nodes of a domain and tunneling the RSVP messages over other part of the domain.

1.3.2Differentiated Services

Diffserv architecture is based on a simple model where traffic entering a network is classified and possibly conditioned at the boundaries of the network, and assigned to behavior aggregates. A single Differentiated Services Code Point (DSCP) identifies each behavior aggregate. Within the core of network, packets are forwarded according to per-hop behavior (PHB) associated with the DSCP. Using those PHBs, several classes of services can be defined using different classification, policing, shaping and scheduling rules.

1.3.2.1 Framework

DiffServ architecture consists of a number of functional elements implemented in network nodes, including a set of PHBs, packet classification functions and traffic conditioning functions including metering, marking, shaping and dropping.

1.3.2.2 DS Domain

A DS domain is a set of DS nodes, which operate according to a common service provisioning policy and has a common set of PHBs, is the entity that provides a coherent set of PHBs in the network domain. It consists of DS

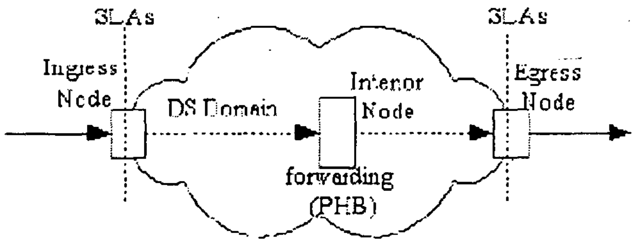


Fig 2. DS Domain

boundary nodes and interior nodes as shown in above in fig 2. DS boundary nodes act both as a DS ingress node and as a DS egress node for different directions of traffic. DS boundary nodes interconnect the DS domain to other DS or non-DS capable domains, while DS interior nodes only connect to other DS interior or boundary nodes within the same DS domain. A boundary node could contain functionality for both logical interconnection of domains and controlling traffic streams. The interior nodes could have certain limited traffic conditioning capabilities, e.g. for degrading the importance level of a packet at each hop. In addition, DS boundary nodes classify and possibly condition the ingress traffic to ensure that incoming packets are appropriately marked to select a PHB from one of the PHB groups supported within the domain. When the packets inside the network are forwarded to their destination, each of the packets gets treated only on per hop basis during the forwarding within the network region. The treatment provided to a DSCP is identical within the DS domain; however the mapping from the DS code point to PHB may be different in different DS domains.

1.3.2.3 DS Region

A DS region is a set of contiguous DS domains. It is capable of offering differentiated services over paths across its DS domains. DS domains in a DS region may support different PHB groups internally and different DSCP to PHB mappings. However, to permit services that span across the domains, each DS domain must establish a service agreement (SLA), which defines a traffic conditioning agreement (TCA). A TCA specifies how transit traffic is conditioned at the boundary between the two DS domains.

1.3.2.4 Traffic Classification

The packet classification policy identifies the subset of traffic, which receives differentiated services by being conditioned and/or mapped to one or more PHB groups within the DS domain. The classification is performed by packet classifiers, which select packets in a traffic stream based on the content of some portion of the packet header. In general, two types of classifiers are widely discussed: the behavior aggregate classifier classifies packets based on DSCP only. The multi-field classifier selects packets based on the value of a combination of one or more header fields, such as source address, DS field, protocol ID, source port and destination port numbers, and some other information.

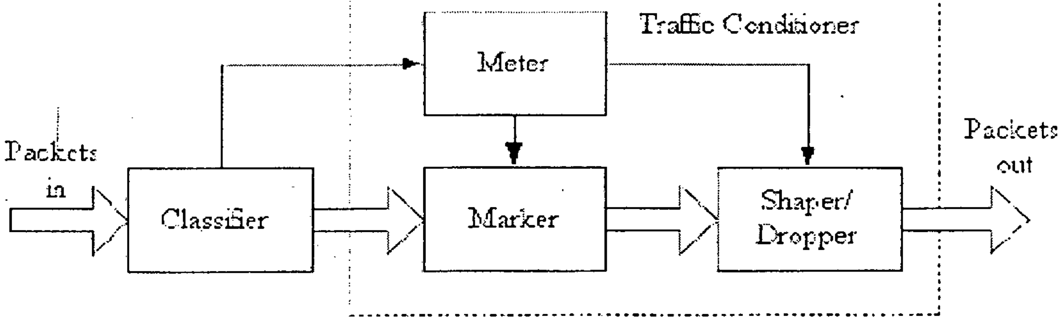


Fig 3. The packet classifier and traffic conditioner

Traffic conditioning is used to enforce rules specified in a traffic conditioning agreement (TCA). A traffic conditioner, which contains meter, marker, shaper and dropper elements, does it. The meter measures each traffic stream and informs the marker, shaper and dropper elements of the status of the stream. The marker sets the importance level of packet according to the given status of the stream. The shaper is used to smooth the traffic stream at a particular aggregate level. Dropper makes discarding decisions based on the content of service level and traffic conditioning agreements.

1.3.2.5 Per-hop Behaviors (PHBs)

The PHB plays a significant role in DiffServ architecture. The term PHB refers to a set of rules that allows for the treatment of packets in a specific way inside the network. PHB defines the forwarding behavior that the packet receives at each hop as it is forwarded through the network. The concept of forwarding behavior can be interpreted to mean those aggregate actions that interior nodes perform with packets having a similar code point in the DS field. PHBs are usually needed in cases when several behavior aggregates (BAs) are competing on resources in a node, which is then able to make service discrimination based on defined PHBs in that DS node.

For a node, the PHB is foremost the means that can be used allocating resources for different behavior aggregates. PHBs can be identified according to how they prioritize the resources, such as buffers and bandwidth or according to how different PHBs are prioritized or how the traffic can be observed in terms of delay and loss. Also PHBs can be used as resource allocation building blocks and thus be grouped together. PHBs are implemented in boundary and interior nodes, usually by means of existing buffer management and scheduling algorithms. Also, in a node, here are possibly more than one PHB implementations that can be grouped together into aggregates or may be separate to each other.

1.3.2.6 DS Code Point Field (DS field)

When mapping a traffic packet to a PHB, the most significant input data is the code point in the DS field of the IP packet header. The description of the DS field octet is shown in fig 4.

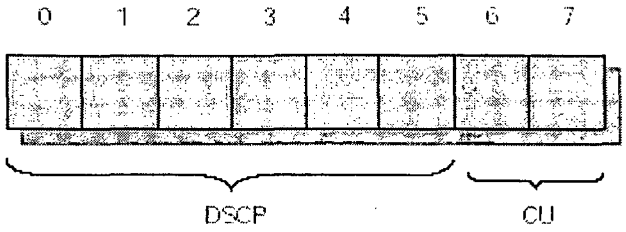


Fig 4. DSCP field layout

As seen, six bits of the DS field are called as the code point (DSCP) and the remaining, which are currently unused, are reserved for the future.

1.3.2.7 PHBs Standardization

With respect to PHBs standardization, the characteristics of PHBs are the subject to be standardized, not the actual algorithms that implement the PHBs. As the DSCP is an octet, there are a total of 64 code points available to be used as PHB standardization. The code point space consists of three pools as follows:

- Pool 1: A pool of 32 recommended PHBs to be assigned to standard actions.
  Bit pattern: 'xxxxx0'.
- Pool 2: A pool of 16 code points to be reserved for experimental or local use.
  Bit pattern: 'xxxx11'.
- Pool 3: A pool of 16 code points that can be used for experimental or local use, but which should be used for standardization in case pool 1 gets overloaded.
  Bit pattern: 'xxxx01'.
  Examples of standardized PHBs:

- **Assured Forwarding (AF) PHB group:** - AF PHB group provides forwarding of IP packets in N independent AF classes, in which there are M different levels of drop precedence. Assured service is intended for customers that need reliable services from their service providers even in times of congestion. Assured service can be implemented by using RIO i.e. a queue with in and out and dropping technique called Random Early Detection [15].

- **Expedited Forwarding (EF) PHB group: -** The EF PHB group can be used for providing services like 'virtual leased line' in which characteristics such as low loss, low latency, low jitter and assured bandwidth play significant role. These kinds of services have two parts:
  - Possibility to configure the DS nodes in a manner that enables minimum departure rate.
  - Policing and shaping the PHB aggregate so that its arrival rate at any DS node is always less than that DS node's configured minimum departure rate.

1.3.2.8 Differences between IntServ and DiffServ

DiffServ is different from IntServ in following aspects: -

- Differentiated service is allocated in the granularity of a class; the amount of state information is proportional to the number of classes rather than the number of flows. DiffServ is therefore more scalable.
- Sophisticated classification, marking, policing and shaping operations are only needed at the boundary of the network. Core routers need only to have behavior aggregate classification. Therefore it is easier to implement and deploy differentiated services.
- InteServ requires ubiquitous implementation, whereas incremental deployment is possible in differentiated services.

1.3.3. Multi-Protocol Label Switching

The MPLS [15] use a fixed length label to decide packet handling. MPLS is also a useful tool for Traffic Engineering. In the OSI seven-layer model, it operates between Layer 2 (L2, link layer) and Layer 3 (L3, network layer). Each MPLS packet has a header. The header contains a 20-bit label, a 3-bit Class of Service (COS) field, a 1-bit label stack indicator and an 8-bit TTL field. The MPLS header is encapsulated between the link layer header and the network layer header. A MPLS capable router, termed Label Switched Router (LSR), examines only the label in forwarding the packet. The network protocol can be IP or any other. This is why it is called Multi-Protocol Label Switching.

MPLS needs a protocol to distribute labels to set up Label Switched Paths (LSPs). MPLS labels can also be piggybacked by routing protocols. A LSP is similar to an ATM Virtual Circuit (VC) and is uni-directional from the sender to the receiver. MPLS LSRs use the protocol to negotiate the semantics of each label, i.e., how to handle a packet with a particular label from the peer. LSP setup can be control driven, i.e., triggered by control traffic such as routing updates. Or, it can be data driven, i.e., triggered by the request of a flow or a Traffic Trunk. In MPLS, a traffic trunk is an aggregation of flows with the same service class that can be put into a LSP. The LSP between two routers can be the same as the L3 hop-by-hop route, or the sender LSR can specify an Explicit Route (ER) for the LSP. The ability to set up ERs is one of the most useful features of MPLS. A forwarding table indexed by labels is constructed as the result of label distribution. Each forwarding table entry specifies how to process packets carrying the indexing label.

Packets are classified and routed at the ingress LSRs of an MPLS-capable domain. MPLS headers are then inserted. When a LSR receives a labeled packet, it will use the label as the index to look up the forwarding table. This is faster than the process of parsing the routing table in search of the longest match done in IP routing. The packet is processed as specified by the forwarding table entry. The outgoing label replaces the incoming label and the packet is switched to the next LSR. This label-switching process is similar to ATM's VCI/VPI processing. Inside a MPLS domain, packet forwarding, classification and QoS service are determined by the labels and the COS fields. This makes core LSRs simple. Before a packet leaves a MPLS domain, its MPLS label is removed. MPLS LSPs can be used as tunnels. After LSPs are set up, a packet's path can be completely determined by the label assigned by the ingress LSR. There is no need to enumerate every intermediate router of the tunnel. Compared to other tunneling mechanisms, MPLS is unique in that it can control the complete path of a packet without explicitly specifying the intermediate routers.

In short, MPLS is strategically significant because:

- It provides faster packet classification and forwarding,
- It provides an efficient tunneling mechanism.

These features, particularly the second one, make MPLS useful[15] for Traffic Engineering.

1.3.4 Constraint- Based Routing & QoS Routing

Network congestion can be caused by uneven distribution of traffic. Some parts of the network are heavily loaded while others parts are lightly loaded. The process of arranging how traffic flows through the network so that congestion caused by uneven network utilization can be avoided is called Traffic Engineering. [15] Constraint based routing is an important tool for making Traffic Engineering automatic. Constraints-based routing is to compute paths for traffic flows with multiple constraints including QoS constraints (requirements) and policy constraints. To determine a path, constraint-based routing considers not only network topology, but also requirements of the flow, resources availability of the links, and possibly other policies specified by the network administrator. Constraint-based routing is recognized as an essential enabling mechanism for a variety of emerging network services such as virtual private networking and QoS support.

In particular, QoS routing, a special case of constraints based routing, which only considers QoS requirements when determining routes, has gained significant importance in the evolution of QoS-based service offerings in the Internet. While determining a route, QoS routing considers not only topology of the network but also requirements of flow and the resource availability of the links. Therefore it may find a longer and lightly loaded path better than heavily loaded shortest path. In order to do this QoS routing needs new metrics to distribute link state information and algorithms to compute routes based on such information. So quality of service routing incurs more overhead cost.

1.3.5 Relation of QoS routing with other techniques

QoS routing is to select the optimal routes for flows so that their QoS requirements are most likely to be met. It is not to replace the Differentiated Services but to help Differentiated Services to be better delivered.

QoS routing determines paths the path for RSVP messages but doest not reserve resources in IntServ. RSVP reserves resources but depends on QoS routing or dynamic routing to determine the path.

MPLS is a forwarding scheme and QoS routing is a routing scheme, MPLS and QoS routing are mutually independent. QoS Routing determines the route between two nodes based on resource information and topology information. It is useful with or without MPLS. Given the routes, MPLS uses its label distribution protocol to set up the LSPs. It does not care whether the routes are determined by QoS routing or by Dynamic

Routing. However, when MPLS and QoS Routing are used together, they make each other more useful. QoS Routing can better compute the routes for setting up LSPs. In combination, MPLS and Constraint Based Routing provide powerful tools for Traffic Engineering.

The following figure shows the position of various techniques in Internet network model:

<table>
<tr><td>Application Layer</td><td></td></tr>
<tr><td>Transport Layer</td><td>Integrated Service RSVP, Differentiated Services</td></tr>
<tr><td rowspan="2">Network Layer</td><td>Constraint Based Routing</td></tr>
<tr><td rowspan="2">MPLS</td></tr>
<tr><td rowspan="2">Link Layer</td></tr>
<tr><td></td></tr>
</table>

Fig 5. Various techniques in network model

1.4 Problem Definition

Routing protocols such as OSPF and BGP currently used on the Internet selects shortest path to a destination, according to metric such as number of hops or that defined by configuration. All traffic for same destination follows the same path (the shortest path), even when there are better paths available according to other parameters e.g. available bandwidth or loss rate. These currently used techniques usually do not take both dynamic state of the network and requirements of traffic into consideration while selecting a path.

In a communication system that aims at providing different quality of service, it is advisable to use routing protocols that make decisions based on the need of the traffic and state of routers in the network. In order to achieve this, it is necessary to distribute new metrics that represent the dynamic state of the network, and use path selection algorithms to compute paths suitable for different types of traffic. However, these benefits come at cost of deploying QoS routing protocols, of incurring potentially higher communication, processing and storage overheads.

In DiffServ model, where traffic is mapped to different classes, there is need for a routing protocol that selects the paths that are adequate to each class of traffic, based on the metric that represent the dynamic state of network from the viewpoint of each traffic class.

In [4] a quality of service routing technique for differentiated services framework is purposed. This technique aims to extend the existing OSPF protocol to broadcast metrics representing the dynamic state of the network. Two metrics called delay index and loss congestion index are used to represent the dynamic state of links of network. Traffic characteristics are expressed using Differentiated Services Code Point (DSCP) field in the packet header.

In this study we evaluate the communication overhead (number of link state advertisements) which directly effects the processing and storage overhead, caused due to dynamic nature of indexes used in above technique, and effect of various combination of mechanisms like thresholds, timers and moving average to control the overhead and their effectiveness to control.

Besides the triggering mechanism, other factors also influence the volume of update traffic. Network topology and connectivity determine the actual number of link state update packets that are flooded on each link. Similarly, the characteristics of requests affect the level of activity in the network, i.e., number and duration of requests, and consequently the frequency of update generation. We attempt to explore the sensitivity of indexes used in above technique to these additional factors by varying network size, topology, and traffic patterns.

Since studying QoS routing in global IP networks is a hard and broad issue, we choose to investigate it in the intranet to simplify the implementation and to evaluate the key factors dominating the feasibility of QoS routing. Further studying such a problem in a real network will be quite expensive and probably may cause some unexpected effects on the network administration. Therefore we choose simulation- based study.

CHAPTER 2

QUALITY OF SERVICE ROUTING

2.1 Overview

QoS routing has been defined in different ways. In [2] it is defined as "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". In [20] it is defined as "a dynamic routing protocol that has expanded it path-selection criteria to include QoS parameters such as available bandwidth, link and end-to-end path utilization, node resource consumption, delay and latency, and induced jitter"

However no matter which definition is concerned, the basic function of QoS routing is to find feasible paths which have sufficient residual resources to satisfy the QoS requirements of the flows while achieving efficiency in network resource utilization. Designing and implementing QoS routing is much more difficult than best-effort routing. Some tradeoffs have to be made. And in most cases, the goal is not to find the best solution, but a solution with acceptable cost.

2.2 Objectives of QoS Routing

Current Internet routing protocols such as OSPF, RIP and BGP are best-effort routing protocols. They use a single objective optimization, which considers only one metric such as bandwidth, hop-count or cost to find a "shortest" path for all traffic. Therefore, even if there are some alternate paths existing, they are not used as long as they are not the "shortest" ones. Obviously, one drawback of this kind of scheme is that it may lead to congestion of some links, while other links are not fully utilized.

In addition, during data transmission, whenever the new shortest path is found, the best-effort routing will shift the traffic to this path from earlier defined shortest path. This traffic may be routed back and forth between different paths. This kind of shift is undesirable because it will bring routing oscillations when routing is based on metrics such as available resources (bandwidth, loss etc.), which are dynamically changing. These oscillations may change rapidly from time to time, and then increase variation in the delay and jitter experienced by the end users. QoS routing is supposed to solve or avoid the problems mentioned above. The main objectives of QoS routing [2] are:

- Improving user satisfaction by increasing chances of finding a path that meets QoS requirements. In case there are several feasible paths available for a given flow, a path is selected dynamically subject to some policy constrains such as path cost etc.
- Improving network utilization by finding alternate paths around congestion spots. This is an objective from service provider's point of view.
- Enabling creation of virtual circuit like services over IP networks.

2.3 Issues Involved

QoS routing differs from best effort routing model of IP network in many ways. Routers running best effort routing protocols exchange routing information which they use to construct their Routing Information Base (RIB), i.e., a collection of routes to all destinations. The routes in the RIB are then used to generate the router's Forwarding Information Base (FIB), which is structured for efficient look-up. The FIB is the data path instantiation of the RIB, which in turn can be viewed as part of the router control path. As packets arrive in the data-path, a destination-based look-up of the FIB determines the outgoing interface on which packets are forwarded. Figure 5 shows a simple functional block diagram for this model, for the case of a link state protocol such as OSPF.

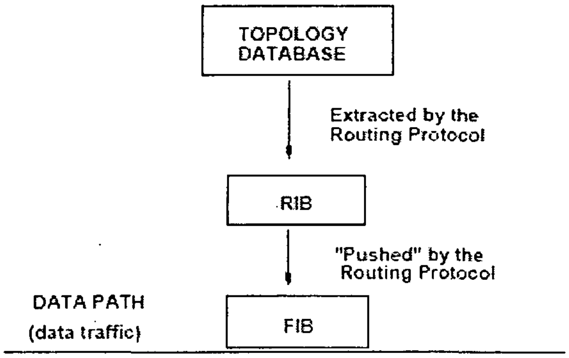
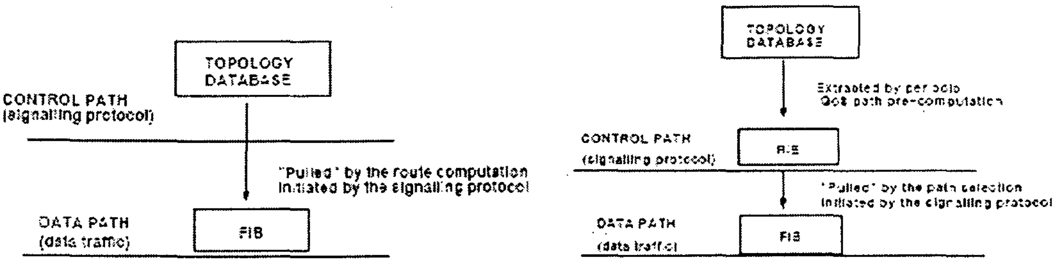


Fig 6. Convntional Router Architecture

The introduction of QoS routing preserves most of the structure of Figure 5, but it introduces a number of small changes. For example, in order to allow determination of paths that satisfy more complex constraints, the RIB now will have to contain more detailed information about the network state, e.g., amount of unreserved bandwidth, delays and so on. This information may be quite dynamic, implying an increase in the amount of control traffic needed to keep the RIBs up to date. Another place where QoS routing departs somewhat from the model of figure 5 is in how the routing information is translated into forwarding state. The main difference is that we can go from a push model used in best effort routing (the routing protocol pushes the content of the RIB into the FIB), to a pull model where QoS routes are selectively inserted in the FIB. The insertion of a QoS route in the FIB is usually triggered when a signaling protocol attempts to establish a QOS path for some traffic.



(a) On-demand path computation (b) Path pre-computation

Fig 7. QoS Router Architecture

The arrival of such a signaling message generates a query to QoS routing to find an appropriate path that meets the traffic requirements. If such a path is found the FIB is suitably modified, so that when packets arrive in the data-path, they get forwarded along the selected path. RSVP and the MPLS path set-up protocol are examples of such signaling protocols.

Figure 7 depicts a simple functional block diagram for this model, where the arrival of a signaling message is the trigger for the creation of the necessary forwarding state (FIB update). Figure 7a shows the case where a path, and hence the associated forwarding state, is computed anew for each request, while figure 7b considers the case where this information is pre-computed and stored in the RIB from where it is then extracted on arrival of the signaling message.

There are benefits to each approach in terms of cost-performance trade-off. The structure of the FIB itself is also affected by QoS routing. Forwarding, instead of being based only on the destination address (or address prefix), can be based on various classification criteria. Reservation signaling protocols define classification criteria for the FIB entries. RSVP, for example, uses a FIB entry based on the 5-tuple (source address, source port, destination address, destination port, protocol number). MPLS, on the other hand, uses labels to determine how to forward data traffic. Each data packet is pre-pended (at appropriate points, e.g., at an ingress router) with a label, and this label is used as the key for looking up the FIB and determining how to forward the packet.

The advantage of a label-based approach is that it can minimize the added complexity in the FIB, To summarize, constraint-based routing, and in particular QoS routing, affects the traditional structure of a router along multiple axis. The data path is impacted because of the need for a larger number of forwarding entries with a different structure than the ones used for best effort routing. Similarly, the control path is also affected, not only because of the additional processing load induced by the signaling messages used to request and set up paths, but also because of the added processing and information exchange required in order to compute QoS routes. The cost of these changes will in turn largely depend on the frequency and granularity of the actions they correspond to, i.e., updates to the RIB and FIB. As a result, there exists a broad range of models and design points for QoS routing that represent different performance goals and operating environments.

2.3.1 QoS Routing for Traffic Engineering

On the one hand, QoS routing is aimed primarily at traffic engineering and its operation is then characterized by a long time scale (long term traffic variations) and a coarse granularity of the traffic flows it handles (traffic aggregates). In such an environment, the goal of QoS routing is maximization of network performance, e.g., minimize delay, in the presence of slowly changing traffic patterns. This is achieved through continuous measurements of traffic patterns and the computation of paths on which to route traffic aggregates so as to optimize various performance measures. One such instance of QoS routing is the optimized-multipath routing. This extends conventional routing protocols by monitoring network performance (most notably utilization of links and delay) and dynamically shifting traffic across pre-computed paths in order to optimize performance. In the context of a traffic engineering application, the different paths computed by QoS routing are either pre-established or change only infrequently, so that when coupled to the long-term nature of the traffic monitoring function, the overall cost of such an approach is typically low. On the other hand, the guarantees that such a traffic engineering based approach can provide are limited. This is because the use of traffic aggregates and the focus on network wide traffic optimization when computing paths make it difficult to provide explicit guarantees to individual flows.

2.3.2 QoS Routing for Dynamic Requests

At the other end of the spectrum of possible QoS routing solutions, QoS routes are computed for each request, where requests explicitly express their resource requirements. Handling of such explicit requests differs from the implicit allocation of resources inferred from the traffic engineering measurements. In particular, requests are likely to be more frequent, and the granularity of resource allocation smaller, e.g., each individual flow may require reservation of resources. As a result, the goal of QoS routing in this environment is also somewhat different. It still targets optimizing network resources, but now under the constraints of satisfying individual request requirements rather than a general measure of network performance. Even through this provides stronger guarantees; it incurs a potentially much higher cost. This is not only due to the greater overhead in updating network state and computing QoS paths for each request, but also due to the signaling overhead to set up individual paths for each request.

While it is possible to design a brand new protocol, there are many basic functions, e.g., neighbor discovery, flooding of updates, etc., that QoS routing shares with traditional best effort, link state routing protocols. As a result, it is typically easier to extend an existing protocol rather than to develop a new one. This is the basis of the proposal [4] that describes extensions to OSPF to support QoS routing.

2.4 Overheads QoS Routing

In this section, we review the core functional components of a QoS routing protocol and their contributions to its cost. These can be broken down into three major categories:

- Protocol Overhead
- Processing requirements
- Storage costs.

QoS routing has effect on each of these factors and cost implications. In addition, issue of the signaling and packet forwarding also introduces some cost in QoS routing.

2.4.1 Protocol Overhead

One basic requirement for supporting QoS routing is to track the availability of network resources, e.g. link bandwidth, delays etc so that this information is available to the path selection algorithm [7]. The Link State Advertisements (LSAs) flooded by OSPF already carry administrative cost metrics for each link, and there is a provision for advertising multiple cost metrics using Type of Service (ToS) fields. As a result, a simple solution is to build on this existing mechanism, and use the ToS fields to encode and flood information such as available link bandwidth and propagation delay. As an added benefit, because the existing OSPF update mechanism triggers the simultaneous flooding of updates for all links on a router, the processing cost of QoS updates will be distributed over multiple links. However, while the distribution of QoS updates can be accomplished with minor modifications to OSPF, additional mechanisms are still needed to determine when updates are to be sent. In particular, routers need to track available resources on their interfaces, and determine when it has changed sufficiently to warrant a new update. The latter is particularly important as it plays a major role in both the protocol overhead and the performance of QoS routing.

The decision of when to flood QoS updates is the responsibility of a triggering function, whose design involves various trade-offs between performance and cost. In particular, a sensitive triggering function that advertises every change in resource level provides the most accurate information for computing paths, but its communication overhead is often not acceptable. A simple method for bounding the communication overhead is to rely on a timer to limit the frequency of advertisements. This clearly provides direct control over the volume of updates, but may not ensure timely propagation of significant changes.

Another alternative is, therefore, to rely on the magnitude of changes as the primary criterion for triggering updates. For example, a threshold based method triggers a new advertisement whenever the change in available resources exceeds a certain percentage of the previously advertised value. Alternatively, resources may be partitioned into ranges or classes, with new advertisements being issued for each class boundary crossing. Such methods provide some control on the trade-offs between information accuracy and volume of updates. However, periods of rapid traffic fluctuations may still trigger frequent flooding of updates and as a result cause transient control overloads, so that threshold or class based triggering functions are often complemented with a hold-down timer to enforce a minimum spacing between consecutive updates.

2.4.2 Processing Requirements

There are several aspects of QoS routing that introduce different or additional processing requirements from traditional best effort routing [7]. The two major ones are: Path Computation and Selection. Path computation is the component whose implementation differs most from its best effort counter-part. QoS routes are computed based on request characteristics, e.g., how much bandwidth is required, and the resource information provided in the topology database. Differences with the best effort model are following: The algorithms used to compute routes, and the conditions that trigger algorithm execution. The latter is a major factor in the computational overhead associated with QoS routing, as well as the in quality of the routes being computed. Paths can either be computed on demand, i.e., for each new request, or precomputed. An on-demand approach has the benefit of being able to always use the most recent information. However, if requests arrive too frequently, this approach may prove costly even if the algorithm is of relatively low complexity. As a result, it is desirable to lower computational complexity. One possibility is to rely on path caching, which seeks to reduce computational complexity by reusing previously computed paths.

Another approach is to pre-compute a QoS routing table in a manner similar to the way a best-effort routing table is pre-computed. However, since the amount of resources requested is not known in advance, such a routing table needs to pre-compute and store multiple alternative paths to each destination, potentially for all possible values of resource requests. In terms of processing load, there are pros and cons with both an on-demand and a pre-computation approach, and either can yield a lower processing load based on operating conditions. Executing the on-demand path selection algorithm is simple since it only involves traversing the topology database and determining a single QoS path. The main contributors to the cost of a single path computation are then the network topology and the relative distance between the destination and the source, although the size of the request and the levels of available bandwidth on network links also have some minor impact on the computational complexity of the algorithm. The more important factor in determining the overall cost of on-demand path computation is the frequency of new requests.

In contrast, path pre-computation is mostly insensitive to the frequency of new requests, and primarily depends on how often the QoS routing table is recomputed, which as opposed to the arrival rate of new requests is a parameter that the router can control. Clearly, frequent re-computations improve accuracy and, therefore, routing performance, but this comes at the cost of a substantial increase in processing load. This is because building a complete QoS routing table is typically more complex than computing a single path, and it further involves the additional cost of de-allocating and re-allocating memory. In addition, when paths are pre-computed, an additional step, i.e., path selection, is required to retrieve a suitable path when an incoming request needs to be routed. A suitable path is one with sufficient resources, and it is retrieved from the QoS routing table by searching column by column the row associated with the selected destination. The search stops at the first entry with available resources larger than the requested value. Note that this means that the cost of path selection will depend to some extent on the requested amount of resources.

2.4.2.1 Processing Link State Updates

Generating and receiving a link state update involves accessing the link state database to extract information or insert newly received information [7]. In addition, the generation of a link state update requires that a packet be assembled and transmitted. Given the greater frequency of updates that QoS routing requires, these are likely to be important contributors to the increased cost of QoS routing.

2.4.3 Storage Costs

There are two areas where QoS routing affects storage costs [7]. The first is the extension of the topology database in order to accommodate link resource availability information. In the context of OSPF, this is easily accomplished through minor modifications to the existing topology database. This extension is facilitated by the fact that resource updates are themselves communicated using existing OSPF mechanisms, i.e., flooding of extended LSAs. As a result, both processing of resource updates and their inclusion in the topology database can be added with minimal modifications.

The QoS routing table itself, when used, also affects storage costs. The size of the QoS routing table depends to a large extent on specific implementation details such as the exact data structures used. However, it is also affected by parameters of QoS routing such as the operation of the triggering policy, which can influence the number of distinct paths that the path computation algorithms generate. For example, there may exist, for each destination, a large number of distinct paths with incrementally different resource values, hence contributing to a larger QoS routing table.

In this dissertation, we ignore the storage cost dimension of QoS routing as it has small impact on the cost. In particular, the increase in the size of the topology database is minor, and although a QoS routing table, when used, can be large, we do not consider it to be a problem for the storage capabilities of modern systems. A detailed discussion of the storage cost of a specific implementation of QoS routing, and its comparison to that of best effort routing, can be found in [14].

2.4.4 Signaling and Packet Forwarding Costs

As discussed above, QoS routing also requires additional forwarding and signaling support, and this does not come for free. The impact on forwarding, i.e., the structure of the FIB, is mostly in terms of the potentially greater number of entries that need to be stored and searched. The added complexity this imposes depends in part on the packet classification approach used in the FIB. The use of a classification, e.g., as with RSVP classifiers, can impose a substantial overhead if it needs to be performed at a very fine granularity. The other and probably more significant cost component is the signaling needed to install the required FIB states in the nodes used by a QoS path. This signaling involves both additional network traffic and processing. The load induced by signaling will clearly depend on the protocol used, but it could be substantial in the presence of a large number of short-lived flows. However, there are various approaches that can be used to mitigate this impact. For example, QoS routing may be reserved for long-lived flows. As a result, while the cost of establishing the reservations that go hand in hand with QoS routing is certainly a topic that requires further investigation, it does not appear to represent a showstopper for QoS routing, i.e., one that would make its deployment completely impractical. In that context, it remains important to assess the other costs and associated complexity of upgrading the (IP) routing infrastructure to support QoS routing.

2.5 Mechanisms for control

The operational cost of the QoS routing depends on the traffic that the network has to handle as well as the parameters of QoS routing protocol i.e. path pre-computation frequency and threshold used for triggering new mechanisms. The followings mechanisms may be used for controlling the control traffic over the network.

- **Timer**: Usually called hold down timer, this parameter provides direct control over the control traffic generated in the network,
- **Threshold:** Threshold value is used to assert that magnitude of resource availability over the links has changed sufficiently to generate link advertisement. More than one threshold values can also be used to provide non-linear variation in resources.
- **Class partitioning**: Resources can be partitioned into classes to generate link advertisement for each class boundary crossing
- **Moving window average:** This mechanism along with threshold can be used to control the communication cost.
- Combination of the two or more is mostly used to provide effective control over the communication cost along with filling the objective of quality of service routing (i.e. making dynamic state of network available)

CHAPTER 3

SIMULATIOR & METHODOLOGY

3.1 Simulator

A simulator using C++ language's object oriented features is written for the purpose of experimentation. The simulator provides the capability of input and output using files. The parameters required for setting the environment are input using files and some parameters are already set to default values. The simulator was designed using various classes implementing nodes and links and interaction among them. A simulation clock was used for timing requirements. All the events are executed using first come first serve method. No error checking capability is provided. The Following figure shows the block diagram of the design of the simulator.

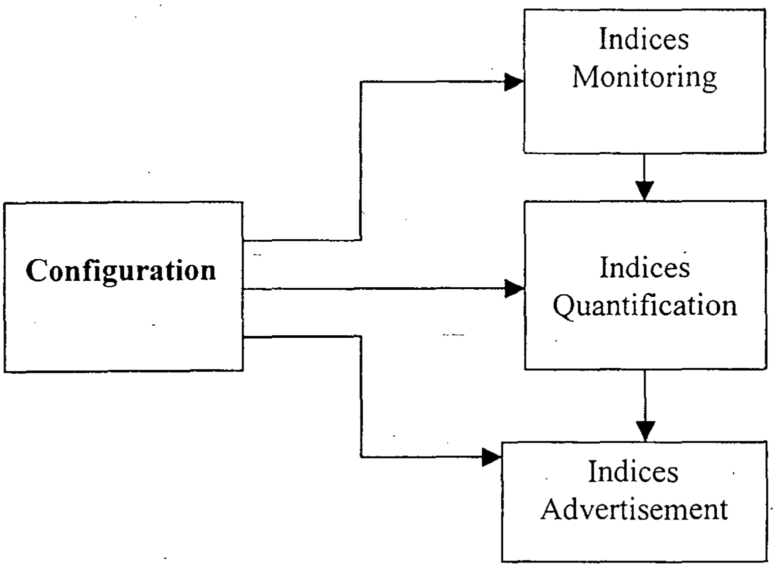


Fig 8. Simulator Block diagram

The simulator provides following capabilities:

- Any number of nodes and links can be specified to set up any typical network topology.

- Moving Average Window size form 1 for immediate previous value, to any positive number for average of more than one previous value can be specified.

- Two threshold values can be specified. Both values can be set to equal. Default value for high threshold is set to half of the lower threshold. This feature is used to provide more sensitivity when index values are less then the average value.

- Three types of traffic patterns can be set up. They are low, medium, and high.

- Any duration can be specified for experiments. Default value is 5 simulation minutes.

- Output of simulator is written to files, which can be analysied using graph drawing facility of EXCEL or xgraph of linux.

3.2 METHODOLOGY

To measure the communication overhead (Number of link state advertisements) broadcast in a unit time and the effect of various parameters on it we have used the following scenario. The emission of advertisements is triggered by a relative criteria consisting of two thresholds high and low. The threshold used determines level of significance, if the value evaluated is below the transition point (Average of defined number of previous values) high threshold is used, otherwise low threshold is used. Figure 9 shows the methodology used in the process.

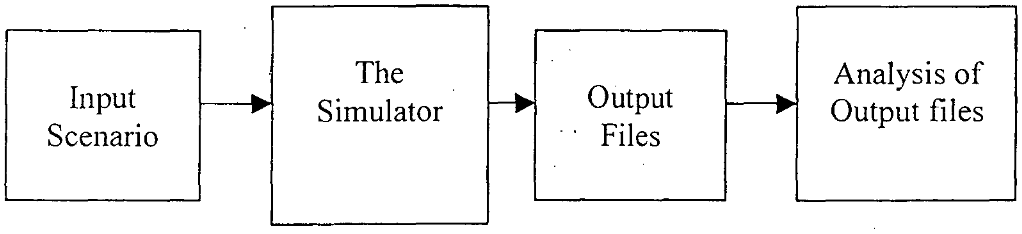


Fig 9. Various Steps of Simulation Process

Four different network topologies, which are typically used in simulating network models, are used. These are shown in the appendix.

- Threshold values are varied from 0 to 100 percent in the increment of 10 percent.
- Moving Average window size is varied as 1,2,5,10,20,30,40,50.
- Simulation time of 5 simulation minutes is used.
- Traffic patterns of low, medium and high are used.
- Graphs of various output data are produced using EXCEL.

CHAPTER 4
SIMULATION RESULTS

The simulation results were evaluated in terms of number of router link state advertisements (R-LSA) issued in unit time in each of the network topologies considered. We only show the graphs concerning the low and high loads because these are sufficient to express the obtained behavior.

4.1 Effect of Threshold

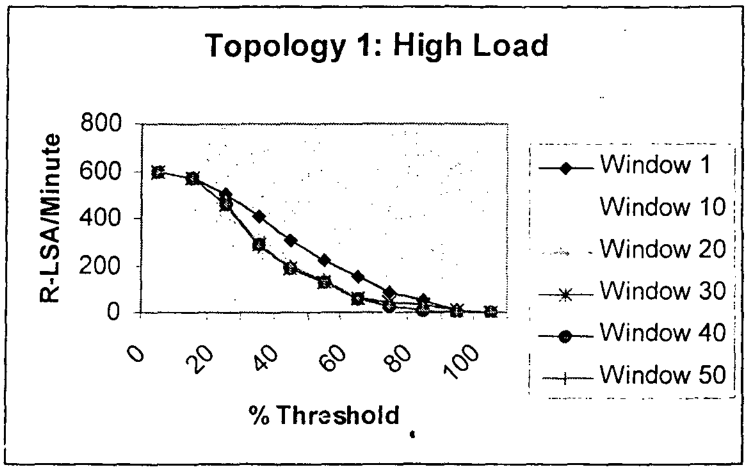
From the graphs it is clear that percentage of the threshold used is effective in controlling the number of link state advertisements issued for most of the topologies. In some of topologies a threshold value of 40% was able to lower the number of R-LSAs by 50 percent.

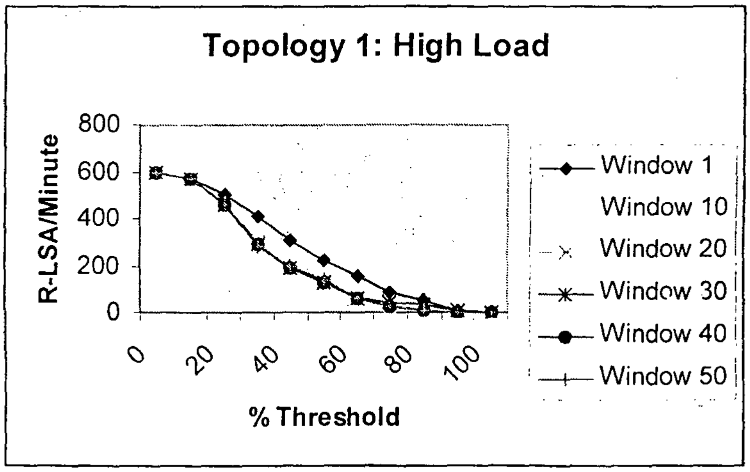
4.2 Effect of averaging window size

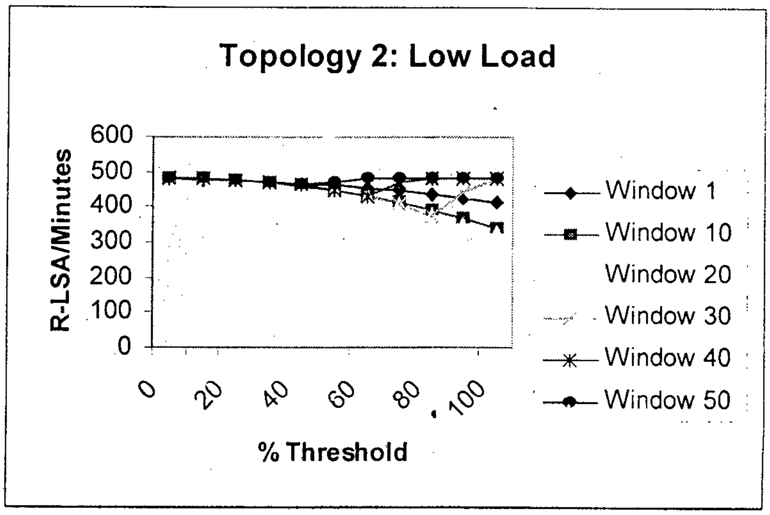
Size of he averaging window is effective in controlling the R-LSAs but subsequent increase in the window size does not improve it further. Window size bigger than 20 marginally improves the R-LSAs.

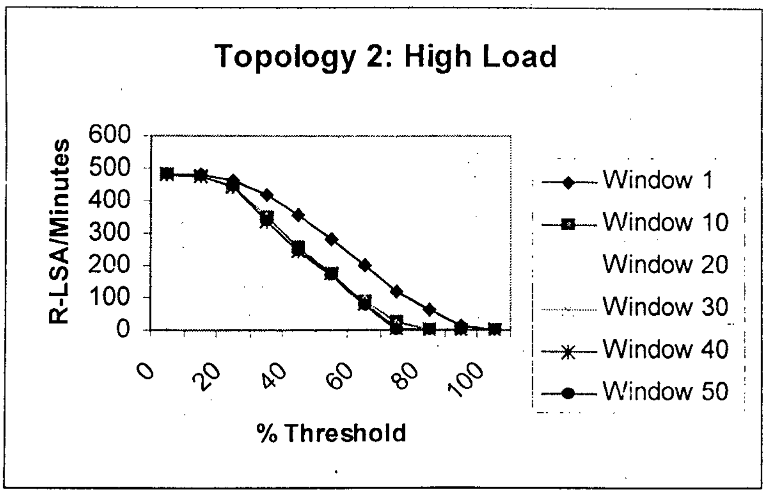
4.3 Effect of network topologies

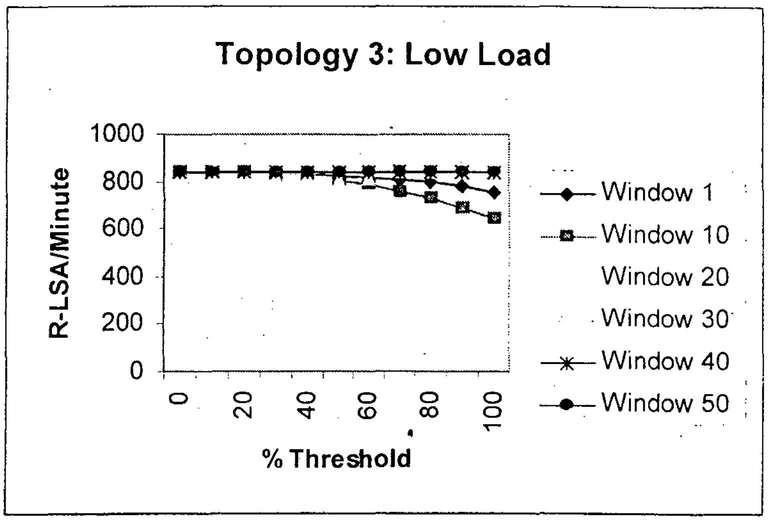
Network topology has substantial impact on the number of R-LSAs. This is probably due to increased number of nodes and links in the network. Window size combined with threshold is effective in controlling the overhead in complex topologies.

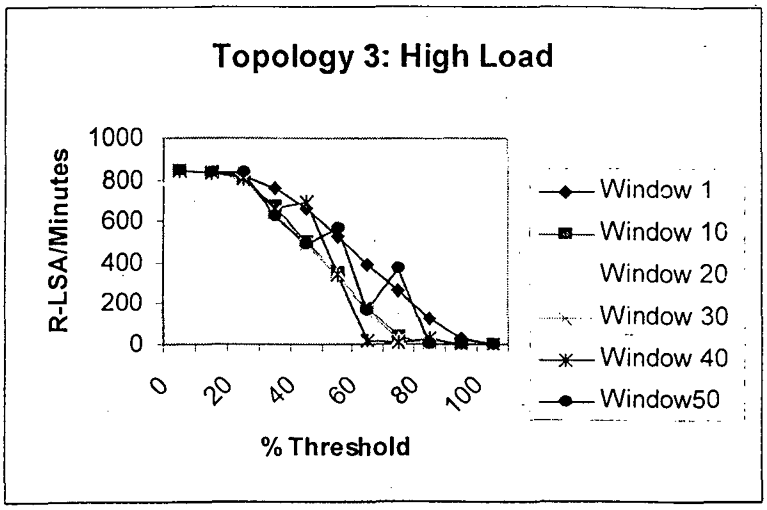


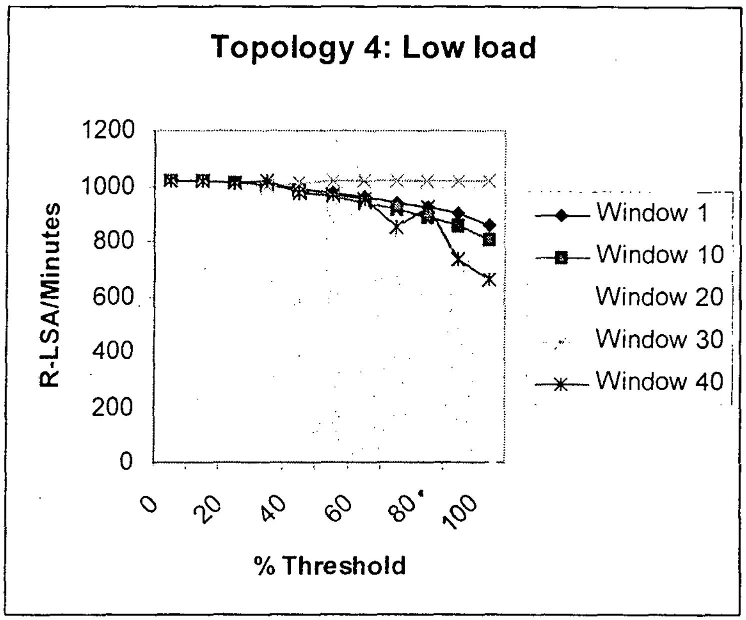


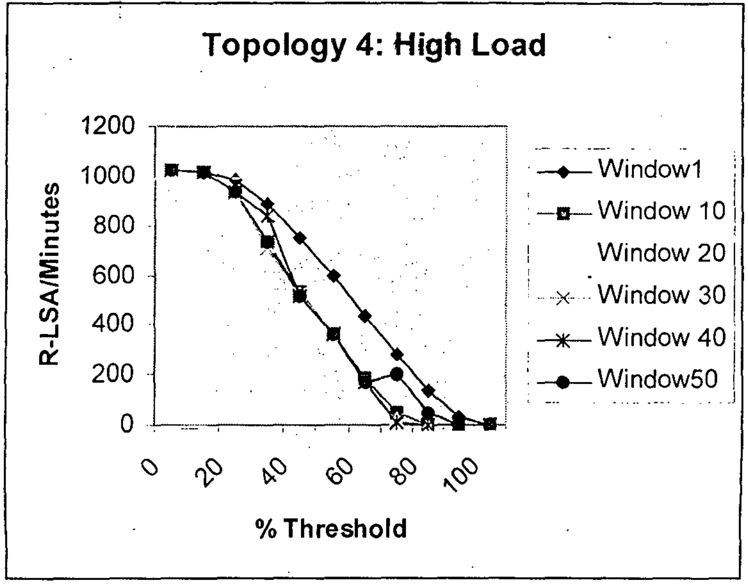

Topology 2: Low Load
R-LSA/Minutes
600
500
400
300
200
100
0
0
20
40
60
80
100
% Threshold
Window 1
Window 10
Window 20
Window 30
Window 40
Window 50


Topology 2: High Load
R-LSA/Minutes
600
500
400
300
200
100
0
0
20
40
60
80
100
% Threshold
Window 1
Window 10
Window 20
Window 30
Window 40
Window 50


Topology 3: Low Load
R-LSA/Minute
1000
800
600
400
200
0
0
20
40
60
80
100
% Threshold
Window 1
Window 10
Window 20
Window 30
Window 40
Window 50


Topology 3: High Load
R-LSA/Minutes
1000
800
600
400
200
0
0
20
40
60
80
100
% Threshold
Window 1
Window 10
Window 20
Window 30
Window 40
Window50


Topology 4: Low load
1200
1000
800
600
400
200
0
R-LSA/Minutes
0
20
40
60
80
100
% Threshold
Window 1
Window 10
Window 20
Window 30
Window 40


Topology 4: High Load
1200
1000
800
600
400
200
0
R-LSA/Minutes
0
20
40
60
80
100
% Threshold
Window1
Window 10
Window 20
Window 30
Window 40
Window50

4.4 Effect of Loads

For higher loads the number of R-LSAs is smaller than with lower loads. This is due to the loss of sensitivity of the threshold mechanism at higher loads. Even with the use of two thresholds the number of R-LSA has decreased substantially.

CONCLUSION & FUTURE WORK

The mechanisms like threshold and moving average are effective in controlling the communication overhead. These mechanisms are comparatively lesser sensitive at higher loads thereby causing fewer number of updates messages. A timer of appropriate value can be used for complex topologies at higher loads to avoid loosing sensitivity. Simulation results show that using adequate moving average window size and threshold values, it is possible to control the communication overhead, and thus showing the scalability feature of the technique.

For future work the following enhancements can be done:

- We have simulated the indices using uniform distributed values. These can evaluated by using various classes of traffic flows with different values.
- The simulation environment used is assumed to be single domain. This can be extended for multiple domains i.e. a DS Region.
- Other aspects like inter-class traffic effects, path calculation overhead, routing stability of the technique can be evaluated.
- Instead of simulation these technique can be evaluated using real network routers and gated software to see real time behaviors.

REFERENCES

[1] William Stallings, "High–Speed networks and Internets performance and Quality of Service", Second Edition, Prentice Hall.

[2] E. Crawley et al. A Framework for QoS-based Routing in the Internet, August 1998. Internet RFC 2386.

[3] J.Moy."OSPF Version 2", IETF, RFC 2328, April 1998

[4] M. Curado, J.Brito, G. Quadros,"Simulation Analysis of UC-QoS Routing Strategy", Proceedings of SCI 2001, Orlando, USA, July 22-25, 2001.

[5] M. Curado, J. Brito, G. Quadros,"Evaluation of a Quality of Service Routing Strategy for the Differentiated Services Framework", Proceedings of the Internet Computing Conference (IC'2001), Nevada, USA, June 25-28, 2001.

[6] J. Heinanen et al.," Internet RFC 2597", Assured Forwarding PHB Group, June 99.

[7] G. Apostolopoulos & S. Kamat," Intra-Domain QoS Routing in IP Networks: A Feasibility and Cost/Benefit Analysis", ICNP'98.

[8] www.qosforum.com, "White Paper QoS Protocols & Architectures".

[9] R.Guerin, D. Williams,"QoS Routing Mechanisms and OSPF Extensions", RFC 2676, IETF, August 1999.

[10] K. Nicols, F. Baker, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers “, IETF, RFC 2474, Dec 1998.

[11] S.Blake, D.Black, M.Carlson, “An architecture for Differentiated Services”, RFC 2475, IETF, December 1999.

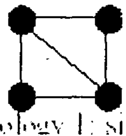
[13] M.Curado, E.Monteiro, “An Overview of Quality of Service Routing Issues”, Proceedings of conference SCI 2001, Orlando USA, July 2001.

[14] G. Apostolopoulos, R. Guerin, and S. Kamat,” Implementation and Performance Measurements of QoS Routing Extensions to OSPF", Proceedings of IEEE INFOCOM'99, New York, March 1999.

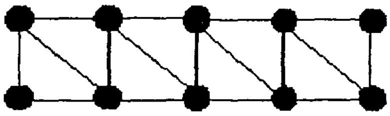
[15] X.Xiao & M. Ni. Lionel,” Internet Qos: A Big Picture “, IEEE NetworkMagazine, March 1999.

[16] QoS forum,”Quality of Service: Glossary of Terms”, May 1999.

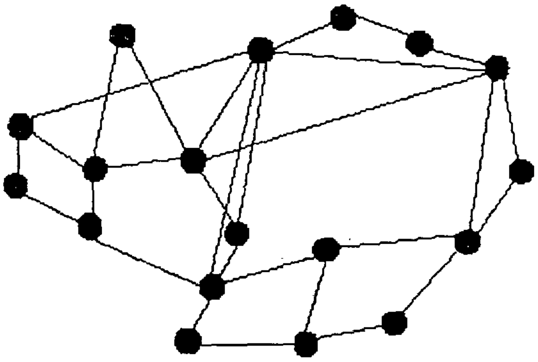
APPENDIX A (Topologies used)



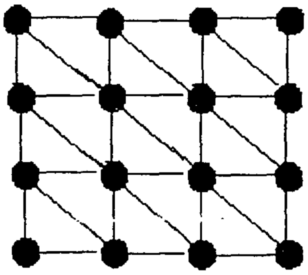
Topology 1: simple network.



Topology 2: network with several alternative paths.



Topology 3. Typical USA ISP backbone



Topology 4 Grid Topology

APPENDIX B (Tables of Results)

Topology 1 Low Load

Threshod (%)	Window 1	Window 10	Window 20	Window 30	Window 40	Window 50
0	239	240	240	240	240	240
10	239	239	238	238	239	239
20	239	237	236	236	236	236
30	236	233	233	233	233	233
40	232	228	227	228	227	228
50	227	221	221	220	221	221
60	223	212	212	211	212	211
70	218	202	199	200	198	199
80	212	189	184	182	183	181
90	208	174	168	165	164	163
100	198	155	150	149	145	143

Topology1 High Load

Threshold (%)	Window 1	Window 10	Window 20	Window 30	Window 40	Window 50
0	600	600	600	600	600	600
10	573	568	567	567	566	566
20	505	464	459	459	460	460
30	412	304	299	291	291	284
40	303	201	190	188	189	192
50	218	134	127	129	129	132
60	152	65	62	60	57	56
70	88	13	44	45	22	25
80	49	11	34	35	12	11
90	10	5	12	12	4	3
100	1	1	1	1	1	1

Topology 2 Low Load

	Window 1	Window 10	Window 20	Window 30	Window 40	Window 50
0	479	480	480	480	480	480
10	478	479	479	479	479	479
20	474	475	476	476	476	476
30	471	469	470	470	472	471
40	466	461	462	463	463	462
50	462	447	448	447	447	467
60	455	433	428	430	430	479
70	447	414	405	405	471	479
80	438	392	380	374	479	479
90	427	367	352	441	479	479
100	411	337	326	479	479	479

Topology 2 High Load

Threshold (%)	Window 1	Window 10	Window 20	Window 30	Window 40	Window 50
0	480	480	480	480	480	480
10	479	475	477	476	476	475
20	462	443	445	444	442	443
30	421	349	341	338	335	336
40	355	254	246	245	246	244
50	279	175	173	173	172	171
60	203	89	86	83	80	78
70	120	22	11	6	5	3
80	60	1	0	0	0	0
90	15	0	0	0	0	0
100	0	0	0	0	0	0

Topology 3 Low Load

Threshold (%)	Window 1	Window 10	Window 20	Window 30	Window 40	Window 50
0	839	840	840	840	840	840
10	839	838	840	839	839	839
20	836	836	835	836	836	837
30	835	829	829	830	836	840
40	829	820	822	832	840	840
50	823	804	808	840	840	840
60	815	784	815	840	840	840
70	806	758	840	840	840	840
80	793	725	840	840	840	840
90	781	688	840	840	840	840
100	756	646	840	840	840	840

Topology 3 High Load

Threshold (%)	Window 1	Window 10	Window 20	Window 30	Window 40	Window 50
0	840	840	840	840	840	840
10	837	837	838	837	837	839
20	818	800	797	800	801	837
30	762	668	640	636	665	626
40	664	498	484	488	693	482
50	529	347	346	341	338	567
60	393	178	174	165	16	162
70	258	41	31	18	11	369
80	129	2	0	0	31	0
90	32	0	0	0	0	0
100	0	0	0	0	0	0

Topology 4 Low Load

Threhold (%)	Window 1	Window 10	Window 20	Window 30	Window 40
0	1019	1019	1020	1020	1020
10	1017	1017	1016	1017	1017
20	1012	1010	1007	1008	1010
30	1002	997	996	996	1015
40	990	982	972	1011	972
50	975	958	951	1019	968
60	961	940	922	1019	954
70	941	918	881	1019	854
80	922	886	840	1019	920
90	900	856	798	1019	735
100	859	805	747	1019	665

Topology 4 High Load

Threshold (%)	Window 1	Window 10	Window 20	Window 30	Window 40	Window 50
0	1020	1020	1020	1020	1020	1020
10	1016	1012	1012	1011	1012	1012
20	983	930	929	931	931	937
30	887	734	714	706	835	731
40	752	534	520	522	514	515
50	594	363	363	361	360	362
60	434	188	169	170	166	171
70	280	46	25	17	10	201
80	134	1	0	0	0	46
90	32	0	0	0	0	0
100	0	0	0	0	0	0