End-to-End Delay Margin Based Traffic Engineering

Mohamed Ashour

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada
2007

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Your file Votre référence
ISBN: 978-0-494-50769-8
Our file Notre référence
ISBN: 978-0-494-50769-8

Abstract

To generate profit and support high volumes of traffic, IP backbone networks are required to provide efficient scalable end-to-end Quality of Service (QoS) for a variety of telecommunication services. Scalability is achieved by using class-based QoS provisioning approaches such as Differentiated Services (DiffServ), while efficiency requires the appropriate use of traffic engineering mechanisms such as those provides by Multi-Protocol Label Switching (MPLS). This thesis presents a delay-margin based Traffic Engineering (TE) approach to provide end-to-end QoS in MPLS networks using DiffServ at the link level. The TE, combines mapping flows to routes, mapping routes to QoS classes, and the definition of each class delay. The thesis formulates traffic engineering as a nonlinear optimization problem that reflects the inter-class and inter-link dependency introduced by end-to-end QoS requirements of flows, and their aggregation into DiffServ classes. Three algorithms are used to provide a solution to the problem: The first two, centralized offline route configuration, and link-class delay assignment, operate in the convex areas of the feasible region to consecutively reduce the objective function using a per-link per-class decomposition of the objective function gradient. The third one is a heuristic that promotes/demotes connections at different links in order to deal with concave areas that may be produced by a trunk route usage of more than one class on a given link. Approximations of the three algorithms suitable for online distributed TE operation are derived. Simulation is used to show that the proposed approach can increase the number of users while maintaining end-to-end QoS requirements.

To estimate the queue length and delay distributions versus changes in the traffic characteristic, the available capacity variation, the loading, and the change in queue weights that are introduced by the proposed TE approach, a multi-scale queue performance analysis technique is proposed. At each scale, the queue weights or priority dependencies are exploited to convert the multi-queue problem into a set of single-queue problems. The core of the analysis is in using Variable Service rate Multi-Scale Queuing (VS-MSQ) to estimate the multi-scale capacity available to each queue. The thesis shows the hierarchy of this estimation and the dependency of the each queue variable capacity on the unused capacity of the other queues and their weights or priority. Simulation and analytical results on the queue length and delay survivor functions are in a good agreement.

Sommaire

Les réseaux fédérateurs IP doivent fournir une Qualité de Service (QdS) bout à bout extensible et efficace à une variété de services de télécommunications. L'extensibilité (scalability) est réalisée par des approches de classification de la QdS telles que la Différenciation de Services (DiffServ). L'efficacité, quant à elle, exige l'utilisation appropriée de mécanismes d'ingénierie du trafic (Traffic Engineering, TE) comme la commutation multiprotocole par étiquette (Multi-Protocol Label Switching, MPLS). Cette thèse présente une approche d'ingénierie du trafic (TE) basée sur la marge du délai pour fournir la QdS de bout en bout dans des réseaux MPLS utilisant DiffServ au niveau de la liaison. Le TE assemble l'attribution des flots d'information aux chemins, l'attribution des chemins aux classes de QdS et finalement la définition de chaque délai de classe. La thèse formule l'ingénierie de trafic comme problème d'optimisation non linéaire qui reflète la dépendance inter-class et inter-liaison produite par les exigences de la QdS de flots bout à bout, et de leur agrégation dans les classes de DiffServ. Trois algorithmes sont employés pour fournir une solution au problème : Les deux premiers, la configuration déconnectée et centralisée d'itinéraire et l'assignation du délai de la classe liaison, fonctionnent dans les zones convexes de la région réalisable pour réduire consécutivement la fonction objectif. Ils utilisent une décomposition par-classe et par-liaison du gradient de la fonction objectif. Le troisième algorithme est un heuristique qui favorise/rétrograde les connexions à différents liens afin de traiter les régions concaves produites par l'utilisation éventuelle du chemin par plus d'une classe sur un même lien. Des approximations appropriées à une opération distribuée en ligne de TE sont dérivées pour les trois algorithmes. La simulation est adoptée pour prouver que l'approche proposée peut augmenter le nombre d'utilisateurs tout en maintenant les conditions bout à bout de la QdS.

Pour estimer la longueur de file d'attente et la distribution des délais par rapport aux changements de caractéristique du trafic, la variation de la capacité disponible, le chargement, et les changements des poids de file d'attente qui sont présentés par l'approche proposée de TE, on propose une technique d'analyse de performance de file d'attente multi-échelle. À chaque échelle

de temps, les poids de file d'attente ou les dépendances de priorités sont exploitées pour convertir le problème de multi-file d'attente en un ensemble de problèmes à simple-file d'attente. Le cœur de cette analyse est l'utilisation de la file d'attente multi-échelle à taux de service variable (Variable Service rate Multi-Scale Queuing, VS-MSQ) pour estimer la capacité multi-échelle disponible à chaque file d'attente. La thèse montre la hiérarchie de cette estimation et la dépendance de chaque capacité variable de file d'attente sur la capacité inutilisée des autres files d'attente et leurs poids ou priorité. La simulation et les résultats analytiques des fonctions de longueur de file d'attente et des survivants de délai sont en accord.

Acknowledgements

I am thankful to God for providing me with the inspirations, the patience, and the strength, needed to finish this thesis.

I am truly grateful to my advisor Professor Tho Le-Ngoc for his exceptional guidance during my PhD studies. Professor Le-Ngoc has always found the time for long and strong research discussion that were instrumental in developing this thesis research work. His careful review of my publications and his strive for high research standards have pushed me to make this thesis contribution stronger. I am also grateful for the professional and academic skills that learned working with Professor Le-Ngoc. These skills are greatly contributing to my Career.

I would like to thank Professor Ioannis Psaromiligkos and Professor Lawrence Chen, for their contribution as members of my Ph.D. committee and providing me with valuable feedback during the phase of my research proposal.

I have been truly lucky to be surrounded by wonderful research group in McGill University. I am truly grateful to have had the chance to do my Ph.D in McGill University. In particular, my most sincere thanks go to my friend and Ph.D colleague Tallal El-Shabrawy. His friendship has helped me go through the difficult times during my Ph.D. I am also thankful to my office mates Ping Wang and Yang Xu for their support and the interesting research discussions especially towards the end of my Ph.D.

My deepest gratitude goes to my wife Marwa Shakeeb. I could not have made it through the Ph.D without her love and support. Special thanks to my parents for their continuous extraordinary support in every aspect of my life. This support has always enabled me to do more and has pushed me to excel. They are a true blessing of God to me.

Table of Content

List of Figures

List of Tables

List of Abbreviations

AF	Assured Forwarding
BE	Best Effort
CR-LDP	Constrained routing Label Distribution Protocols
DiffServ	Differentiated Service
DSL	Digital Subscriber Line
EF	Expatiated Forwarding
FCFS	First Come First Serve
FTP	File Transfer Protocol
IP	Internet Protocol
IntServ	Integrated Service
LAN	Local Area Network
LDP	Label Distribution Protocol
LRD	Long Range Dependent
LSP	Label Switched Path
MPLS	Multi-Protocol Label switching
MSQ	Multi-Scale Queuing
OSPF	Open Shortest Path First
PHB	Per-Hop Behaviour
PQ	Priority Queue
PSTN	Public Switching Telephone Network

QoS	Quality of Service
QC-VS-MSQ	Queue Couples Variable Service Rate Multi-Scale Queuing
RSVP	ReSerVation Protocols
SRD	Short Range Dependent
TE	Traffic Engineering
VoIP	Voice over IP
VPN	Virtual Private network
VS-MSQ	Variable Service Rate Multi-Scale Queuing
WAN	Wide Area Network
WFQ	Weighted Fair Queue

List of Symbols

V	The set of network nodes
E	The a set of directional links
e	Network Link where $e \in E$
G	A directed graph of N of nodes and E of links $G = (N, E)$
s_e	Source node of link e, where $s_e \in N$
t_e	Destination nodes of link e, respectively $t_e \in N$
V	The set of QoS classes
v	link QoS class, where $v \in V$
C_e	Total Capacity of link e.
$\lambda_{e,v}$	Average class v traffic on link e
$d_{e,i}$	Average class v delay on link e
λ_e	Average traffic on link e
d_e	Average delay on link e
M	The set of end-to-end *trunks*
i	An end-to-end network trunk where, $i \in I$
$\hat{s}_i$	Trunk i source
$\hat{t}_i$	Trunk i destination
γ_i	Average traffic requirement of trunk i
R_i	The set of all possible *routes* for trunk i
$\hat{D}_i$	The end-to-end delay requirement of trunk i
P_i	The set of routes used by trunk i

p_i	A route used by trunk i where $p_i \in P_i$
$\gamma_{i,p} > 0$	The average traffic transported by route p_i.
$E_{i,p}$	The set defining route p_i, where element in $E_{i,p}$ represents a link-class pair $\{e, v\}$ that represents a QoS class v in link e.
$\hat{d}_{i,p}$	The average end-to-end delay of route set $E_{i,p}$
$U_{i,p}$	*delay-margin penalty* function of route set $E_{i,p}$
U	Network delay-margin penalty
$L_{n,q}$	Link Length
$\mathbb{P}(event)$	probability of an event
Q	Queue Length
τ	Time scale
τ_m	Time scale m
$K[\tau]$	Total arrival in the time interval τ
$C[\tau]$	Total service in the time interval τ
$A_m^{(K)}$	Multi-scale Beta distributed multiplier of the arrival K at scale m
$A_{*m}^{(K)}$	Beta distributed characterization the arrival K at scale m
$\beta(a, b)$	Beta function with parameters a and b
$f_\beta(x;a,b)$	The *pdf* of a beta random variable with parameters a and b evaluated at x.
$\beta'(a, b, c)$	Incomplete Beta function with parameters a and b evaluated at c
$f_{\beta'}(x;a,b,c)$	The *pdf* of an incomplete beta random variable with parameters a and b evaluated at .
$f_X(y)$	*pdf* of the random variable X evaluated at y
w_v	Weight of QoS class queue v
$\hat{C}_v$	Modified service rate when queue decoupling is considered.
	Queue State
$R_v[\tau]$	Remaining capacity form queue v in the time interval τ
$R_{v1\|\{v2,v3....\}}$	Second level of remaining service rate, that defined as the remaining service rate from queue $v1$ given that it is already using the remaining service rate form queues $v2$, $v3$,......

Chapter 1

End-to-End QoS Provisioning in Backbone Networks

1.1. Backbone Network

A backbone network is the set of connections and networking devices used to interconnect a number of edge networks. Using the backbone network, users on different edge networks connect to each other, and to application servers on other networks. As shown in Figure 1-1, edge networks can vary from internet access networks such as Digital Subscriber Lines (DSL) and high-speed internet cable networks to more telecommunication oriented networks such as cellular networks, legacy telephone networks and cable TV networks. Communication services provided to the end-users are a mix of internet protocol based services, legacy telecommunication services, and newer services that are constantly emerging. These edge networks present the backbone network with high volumes of traffic, and a big variation of Quality of Service requirements. A backbone network is usually owned by a service provider that is interested in generating profit. Generating profit in backbone networks requires maximizing the number of edge network connections (admitted to the network) without increasing the network resources. On the other hand, gaining customer satisfaction requires guaranteeing the QoS requirements of accepted connections. For the same set of network resources, maintaining the QoS requirements of each user becomes more difficult as the number of network users increases. Maximizing the number of network users while guaranteeing QoS requirements is challenging in current backbone networks.

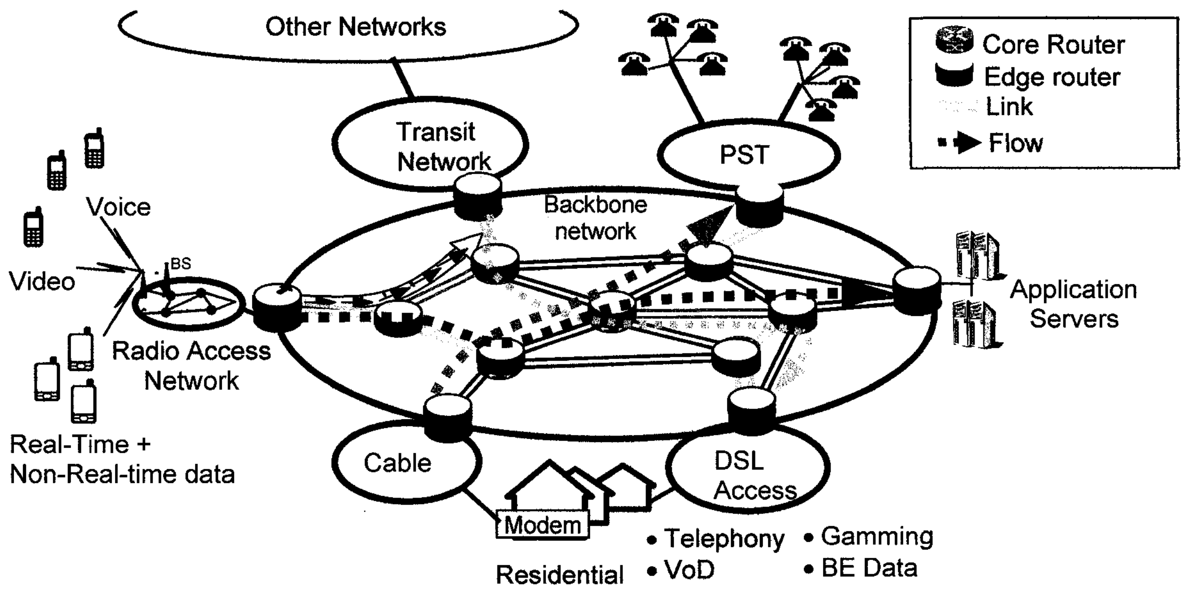


Figure 1-1: Example core IP network

1.1.1. Backbone network challenges

Traditionally a backbone network supported one telecommunication service. Having a separate network for each service allowed service providers to choose the backbone technology that is most capable of satisfying the QoS of the service they offer. Telephone service providers used a centrally managed circuit switched backbone to support the voice tight delay and packet loss requirements. On the other hand, data networks used a distributed packet switching approach, because connectivity and throughput efficiency were originally more important than QoS in data networks. The problem with maintaining a separate backbone network for each service is that it increases cost and prevents the sharing of unused resources. In order to reduce their cost, edge network providers are interested in using one network technology to support all their services. Each of these services has a different set of end-to-end QoS requirements. While treating all services equally is simple, it is inefficient. Treating all services equally requires operating at low load to guarantee the most stringent QoS requirement. To support different QoS requirements without sacrificing efficiency, the network have to employ efficient QoS provisioning and traffic engineering approaches.

The high volumes of traffic associated with backbone networks require the use of high bandwidth links and fast routers. The implementation of fast routes requires the use of simple routing control algorithms. Using scalable approaches is highly desirable in backbone networks. It allows the network to cope with the number of end-user connections that can vary from several hundreds to millions of connections in each router. Scalable algorithms have the advantage of presenting minimum increase in the network devices processing, complexity, and memory requirements versus the increase in the number of these users. Scalability can only be achieved through the aggregation of end user connections and distributing the network functionality over all network nodes. Aggregation increases the statistical multiplexing gain, but it decreases the granularity of end-to-end QoS. Coarse end-to-end QoS choices cause inefficient use of resources. On the other hand, allowing nodes to take local decision in a distributed approach requires less processing and keeps less state information, but it decreases the system ability to achieve optimal use of network resources. Using a scalable distributed QoS provisioning and traffic engineering approaches that optimally utilize network resources to guarantee accurate end-to-end QoS presents a challenge for backbone networks.

QoS provisioning and traffic engineering require the availability of analytical models to estimate the network performance. Legacy voice and Best Effort data depended on models that assume exponential call and packet inter-arrivals. The recently discovered self-similarity and long-range dependent (LRD) behaviors of Internet traffic (e.g., [1]-[3]) pose another challenge in providing efficient and scalable end-to-end QoS. It makes performance evaluation and network optimization algorithms based on exponential models [1] very optimistic. It also makes resource reservation and network traffic engineering (TE)[1] algorithms based on the effective–bandwidth approach [4] very inefficient [5]. Supporting multiple services requires using multi-queue QoS scheduling approaches. Analyzing the performance of these queues in the presence of LRD traffic is challenging, but essential to provide efficient QoS provisioning and in traffic engineering the network.

1.1.2. IP backbone networks

In recent years, there has been a steady move towards using Internet Protocol (IP) routers and switches in building backbone networks [6]-[7]. IP promises a reasonable solution for some of the backbone network challenges. The simplicity and the packet switching nature of internet protocols provide a low cost solution to building backbone networks. The ability of IP to encapsulate data from different communication services into its variable size packets allowed using IP to support multiple services. IP packet switching reduces cost by allowing more users to share the network bandwidth. The simplicity and distributed nature of the IP Best Effort (BE) approach reduce the complexity of data forwarding, hence allowing the use of low cost devices to support a large number of connections. With these advantages comes the inefficiency of IP best effort in providing end-to-end QoS.

Internet protocols are designed to provide connectivity. The IP Best Effort main objective is to reach the destination even if transmission quality is poor, and everybody is expected to receive the same service. Although IP allows for implementing multi-class queuing based on the IP packet packet Type of Service (ToS) field, most router implementations used a single First Come First Serve (FCFS) queue to schedule all traffic heading to the same output link. The use of a FCFS queue makes it impossible to provide different QoS to different flows. BE routers are relatively simple. The majority of IP routers used the Dijkstra algorithm to select the shortest

[1] Traffic engineering (TE) [21] is defined as part of the Internet network engineering dealing with the process of controlling how traffic flows thought the network in order to optimize resource utilization and network performance.

path to a given destination. BE routing forwarded all the traffic heading to the same destination along the same shortest path, independent of the loading or the current link QoS. Shortest path can not guarantee the route QoS because it has no means to limit the route search to routes that satisfy a certain QoS requirement. BE has a very limited traffic engineering capability because of its inability to map same destination connections to multiple routes. Although OSPF and Intermediate System-to-Intermediate System (IS-IS) explicitly allow Equal-Cost Multi-path routing [8], they only provide multiple forwarding entries at each vector routing table. The use of vector routing limits this traffic engineering capability to per-packet multipath routing over equally short routes. This per-packet multipath routing did not provide means to define end-to-end multiple paths or define or to adjust the loading of each path. Providing efficient end-to-end QoS has two main ingredients: QoS provisioning; and QoS Traffic Engineering. IP best effort faces challenges in both ingredients.

1.1.3. IP QoS provisioning

The advance towards IP QoS provisioning used either a flow or a class-based approach. The flow-based approach (e.g., IntServ [9]) can provide a very accurate end-to-end QoS, because it uses a separate queue to schedule each flow. This poses a scalability problem, as flow-based approaches have to monitor and keep information about each flow. In class-based approach (e.g. DiffServ), scalability is improved by grouping flows at each link into a set of classes. A separate queue is used to schedule each class traffic at the output link. The end-to-end QoS provided by DiffServ is the sum of the QoS provided by scheduling at each node. Although services mapped to the DiffServ classes have defined end-to-end QoS requirements, the QoS experienced by a given class between any two network edges will vary. This variation will depend on the route length between any given two nodes and the local QoS provided at each node. The local QoS provided by the scheduler at each node can be relative (e.g., [10],[11] and[12]) or absolute (e.g., [13],[14],[15],[17], [19]). Relative techniques provide vague end-to-end QoS. Ensuring that all the end-to-end services remain below their requirements, requires the use of end-to-end QoS management techniques such as rerouting, rate control and connection admission. Absolute class-based QoS provisioning can guarantee an end-to-end QoS that is a function of the QoS provided by the route links. However, it presents the problem of defining the local QoS requirements such that the end-to-end QoS requirements of all services are satisfied. This thesis, aims at providing an end-to-end QoS using absolute class-based QoS in order to maintain network scalability.

Knowing the QoS provisioning schemes at each link, routing algorithms are needed to search for the route that can satisfy the end-to-end QoS at the minimum possible use of network resources (considered as cost). Several research proposals have been introduced for routing a single flow constrained by QoS requirements. A survey of these techniques is presented in [16]. QoS routing chooses routes to achieve the optimal flow at the flow establishment instance. These route choices may not remain optimal upon the arrival of other flows. Another choice is to consider the global situation in order to find an optimal configuration for all the routes in the network as part of the network Traffic Engineering (TE).

This thesis focuses on traffic engineering as a means to enhance the network utilization and to increase the QoS satisfaction. More specifically, it addresses the QoS constrained multi-class traffic-engineering problem. [18]

1.2. Traffic Engineering

TE applies business goals, technology, and scientific principles to measurement, characterization, modeling, and control of internet traffic in order to achieve global network performance objectives [20]. Unlike network engineering where the objective is to plan the optimal addition of network resources, TE [21] works on a network wide level to find a network configuration that optimally utilizes the existing network resources. In the class-based QoS provisioning used in this thesis, the network configuration determines the assignment of traffic to routes, the mapping of traffic to QoS classes, the amount of resource reservation, and the tuning of traffic schedulers.

1.2.1. IP Traffic Engineering

Service providers in BE use either planning or link metric adjustments to optimize the use of their networks. Planning either increased the Band Width (BW) of links, or added more link connections to overcome the increase in loads. Changes of the link length metric were used as a means to changing the routing configuration ([22]-[24]). Increasing the length of a given link pushed routes going through it to be non-shortest path and hence deflected traffic away form this link.

The introduction of IntServ [9] added traffic engineering capability by providing path establishment and resource reservation mechanisms using RSVP ([25],[26]). However, the network ability to distribute traffic in IntServ remained limited. On the other hand, DiffServ only focused on per node QoS provisioning and had no routing or traffic engineering capabilities inside the network. Traffic engineering in DiffServ networks depends on Bandwidth Brokers (BB) that control access to the network in order to maintain Service Level Agreements (SLAs) between DiffServ networks.

1.2.2. Multi Protocols Label Switching (MPLS)

The introduction of Multi-protocol Label Switching MPLS [29] provided IP with traffic engineering capabilities. MPLS provide fast and simple forwarding of backbone network traffic by establishing virtual paths known as Label switched Paths (LSP) between Label Edge Routers (LER). LSPs are established by setting incoming and outgoing label bindings at each Label Switched Router (LSR) along the LSP. Outgoing labels indicate the forwarding direction (link) as well as forwarding behavior (DiffServ class) to be applied to that traffic. The use of labels gives MPLS the ability to choose a different path for each flow. LSRs can swap, add or drop a label. Adding a similar label to packets from different flows, forces the network to treat them as one aggregate in consecutive nodes. On the other hand, adding different labels to packets from the same LSP splits the flow to a number of smaller flows. The add/remove labels gives MPLS distinct traffic engineering capabilities. The aggregation allows for increasing network scalability, while splitting allows for distributing the traffic on different network links. Details about MPLS architecture and characteristics are discussed in [29]. An overview of the traffic engineering techniques associated with MPLS can be found in [20].

The MPLS architecture focused on simplifying data forwarding and enhancing the traffic engineering capability, and left open the per-node QoS provisioning. MPLS initially assumed a flow-based approach where RSVP-TE or CR-LDP[28] protocols are used to reserve resources on a per-flow bases. To increase the scalability, using a class-based approach provided an attractive solution.

1.2.3. MPLS-DiffServ

The MPLS-DiffServ architecture [29] presented the first step towards introducing class-based QoS functionality in MPLS. However, the architecture focused on using MPLS LSPs to connect DiffServ networks. The objective was to allow MPLS to maintain the differential relation when used to connect DiffServ networks. The MPLS-DiffServ architecture [29] supported two approaches E-LSP and L-LSP. In both MPLS-DiffServ approaches, the MPLS header determines the outgoing links. To define the LSP DiffServ PHB, E-LSPs only use the experimental bits in the MPLS header, while L-LSPs use both the MPLS label and the experimental bits. Based on the inferred PHB, a DiffServ scheduler is used to share resources reserved for the E-LSP or the number of L-LSPs used to connect two DiffServ networks.

Another alternative would be to use DiffServ for the local QoS provisioning in MPLS networks. In this approach LSPs are forwarded based on their labels, but are scheduled at the output link according to their PHB. MPLS in this case provides the forwarding, the traffic aggregation and splitting, while DiffServ schedules traffic at each link. LSPs with the same PHB are input into the same queue. This increases the scalability of MPLS by allowing each link to maintain a number of queues equal to the number of QoS classes instead of the number of LSPs.

1.3. Class-Based Traffic Engineering Considerations

Traffic engineering has been a well-established research field especially in circuit switching networks. Although the MPLS-DiffServ TE introduced in [27] aims to provide traffic engineering for the MPLS and DiffServ combination, it only focuses on per link issues. The TE functionalities in [27] provides means to limit each class link share, to ensure that each class traffic matches the scheduler configuration, and to ensure guaranteed services will delivered within the tolerance of their service agreements. However the traffic engineering in [27] did not provide means to guarantee end-to-end QoS using MPLS and DiffServ. It aims to maximize the efficiency of the network resource utility at minimum QoS violation. Traffic engineering was mostly concerned about choosing a set of routes (routing configuration) that optimizes a certain network metric. Single class traffic engineering used network optimization metrics such as average path length, total residual bandwidth, minimum maximum interference, minimum average path delay, and minimum average path packet loss. The basic route configuration problem can be formulated as a multi-commodity flow optimization problem. The basic formulation of this problem is NP hard [30]. The problem complexity is reduced by decomposing the problem into a set of sub-problems. Although QoS was included in the traffic engineering by using path delays as the optimization metric in load balancing (e.g. [30][31]), these approaches aim to minimize the network overall delay. However, when flows have end-to-end QoS constraints, the optimal combination of delays obtained using load balancing may violate the end-to-end requirements of some flows. Adding the end-to-end QoS constraints causes the problem to become NP hard again. Although approaches exist for decomposing (partitioning) this end-to-end QoS into local QoS constraints (e.g. [37], [38], [40]), the decomposition itself is another optimization problem that is associated with each link.

The use of class-based QoS provisioning in this thesis requires combining the choice of QoS class assignment with the choice of a routing configuration. The limited number of QoS choices and the aggregation associated with class-based QoS provisioning add another dimension to the problem. The limited choices may cause an end-to-end flow to receive a QoS that is much better than its requirement and hence making the network ineffective by reserving more bandwidth. On the other hand it may be more efficient to add a flow to higher class aggregate because that class will present it with better statistical multiplexing and hence better link efficiency.

Although the multi-class network can be modeled as a number of separate networks, one for each class, these networks are not independent. Links on these networks do not have fixed capacity. Adding a flow to a given class causes the capacity on that class to increase and decreases the share of other classes. To increase network efficiency, class-based approaches such as DiffServ use work conserving queue scheduling approaches such as Priority Queuing (PQ), and Weighted Fair Queuing (WFQ). Work conserving schedulers distribute resources between different classes such that unused resources from inactive classes are redistributed among classes that have traffic to send.

The problem with work conserving schedulers is that they introduce dependency between the queues. Adding traffic to one class does not only affect the performance of that class queue, but also affects the performance of other queues. Consider the example of a priority queue, where adding traffic to the higher priority class degrades the performance of both higher and lower priority classes. This dependency introduces two main problems to traffic engineering. Firstly, it complicates the choice of routes and their assignment to QoS class. Secondly, it renders the performance estimation associated with traffic engineering more complex. A multi-class QoS routing approach has to consider the interdependency between classes. The effect of routing traffic along a given class on the end-to-end QoS experienced by other classes has to be considered by the QoS routing approach. A given route may be the best choice from a single class point of view; however the QoS degradation to other routes caused by this choice can degrade the overall QoS seen in the network. It would be desirable in multi-class networks to choose the route that would cause the least QoS degradation for all routes in the network without increasing the complexity of the system. Analyzing the performance effect of routing a traffic through a given link class requires decoupling the multi-queue scheduler into a number of FCFS scheduler. The decoupling becomes more challenging in the presence of self-similar traffic. In order to enable providing end-to-end QoS, this thesis has to deal with the following issues.

1.3.1. Target parameters

Traffic engineering approaches have mostly focused on minimizing the average network delay. Minimizing average network delay in the case of Poisson traffic has the effect of balancing the load in the network. Load balancing has specifically received a lot of interest since it acts as a preventive measure against congestion by increasing the amount of unused bandwidth [30]-[36]. When QoS is not considered, this unused capacity can be utilized to support more users in the network. In QoS based networks, the network has to employ capacity reservation to guarantee the end-to-end QoS. A route end-to-end QoS requirement maps to a set of link capacity reservations across the route. On each link (on the route), the capacity reservation translates to a maximum loading that the link can not exceed. As the link loadings approach their maximum allowed loading, the route end-to-end QoS approaches the requirement. This means that the positive margin between the QoS experienced by the flow and the QoS requirement can provide more information about the ability of the router to accommodate more QoS traffic. In this thesis we use delay as the QoS metric and propose a traffic engineering approach that aims to maximize the end-to-end delay margins of network flows. Using delay margin as the target parameter poses the following questions:

- *How to construct a delay margin based route optimization problem?*
- *How to explicitly include the end-to-end QoS constraints in the route configuration?*
- *Can a delay margin based route optimization be decomposed to a per-link length function?*

1.3.2. Mapping traffic into classes

At each node, the traffic engineering has to decide the assignment of flows to QoS classes. Assigning a flow to a given class changes the class traffic characteristic and the delay operating points that have to be maintained by different links. Delay operating points define the absolute QoS limits that have to be maintained by link schedulers. Aggregating flows into classes increases efficiency, however the flow to class mapping associated with DiffServ decreases this efficiency. DiffServ uses service-based differentiation, which maps traffic flows to PHBs based on the applications supported by the flow. For example, DiffServ maps voice sessions to the same forwarding category. The service-based mapping has two problems. First, as the number of services increase, this approach will produce a very wide range of QoS classes. Secondly, flows from the same service can present different per node QoS requirements. Although same service flows require the same end-to-end QoS requirement, the route length, and the loading of different links determine the per-link QoS requirements. A long route requires stricter per-link QoS because the end-to-end QoS is distributed over a bigger number of links. For the same end-to-end QoS, links along a route with a smaller number of links can tolerate worse per-link QoS. This means that the same DiffServ class is a service-based mapping can be presented with a wide variety of local QoS requirements. To guarantee the end-to-end QoS of all flows, each link class will have to provide the highest QoS of all the traffic flows mapped to the class. Guaranteeing higher QoS requires using lower class loadings. Using lower loading for high volumes of traffic wastes more bandwidth. The same problem happens for flows that go through congested links. Flows going through a congested link would experience low QoS and hence would require strict QoS requirements at subsequent nodes compared with flows that go through unloaded links.

This thesis aims to provide accurate end-to-end QoS using a class-based approach. The thesis uses absolute QoS to increase the accuracy of end-to-end QoS. One of the steps towards achieving this objective is looking for an alternative mapping that would provide a better link utilization and that would be more suitable for absolute class-based provisioning. In this case, it is desirable to use a QoS based mapping. The objective of this mapping would be to provide end-to-end connections with the best possible QoS. Assigning the best possible QoS will ensure the satisfaction of the flow QoS requirement; however, it may lead to inefficient use of the network resources. In a network with high load, the mapping has to ensure that flows are assigned to higher QoS classes only when they need it. Because assigning traffic to higher QoS classes degrades the QoS of lower classes, the mapping approach has to ensure that mapping a flow to a QoS class will not cause QoS violations to other classes. The mapping will affect the QoS received by each route and hence the problem of proper class assignment and the problem of finding the appropriate route configuration cannot be considered separately. This poses the following questions:

- *Can a delay-margin based traffic engineering combine both routing and mapping flows to QoS classes?*
    - *Can the assignment of flows to QoS classes be optimized in way that will ensure satisfying the QoS of all flows?*
    - *What is the effect of moving a flow from one QoS class to the other?*
    - *How to include the effect of mapping a flow to a class on the QoS received by other classes in the optimization processes?*

1.3.3. Reservation and QoS partitioning

Resource reservation is the set of approaches and algorithms that define how much resource will be reserved for each connection, and how to reserve them. Resource reservation can be explicit or implicit. Explicit resource reservation reserves at each link the amount of resources needed to maintain the connection. Implicit resource reservation keeps track of the network resources and prevents the addition of connections requiring more than the available resources. Reservation can also be classified as soft, or hard. Hard reserved resources of a given connection cannot be shared by other connections. Soft reservation allows the sharing of resources assigned to a given connection, but ensures with a given probability that the connections will get its resource share when it needs it.

In flow-based approaches, resource reservations were either based on traffic shaping at the edge, or partitioning the end-to-end QoS. Traffic shaping [41] smoothes the traffic at the network ingress to an effective bandwidth, and reserves only the shaped traffic effective bandwidth in core nodes. In this case, most of the queuing delay is incurred at the edge node, and very few queuing resources are required in intermediate nodes. The effective bandwidth is calculated such that the delay at the edge node does not exceed the end-to-end delay. Inside the network, rate-based reservation [42] has the advantage of converting the end-to-end QoS constraint to a bit rate constraint that has to be satisfied on all links along the route. Using these link rate constraints, QoS routing simplifies to the problem of finding the route where the bottleneck link can accommodate the flow rate [43]. The problem with rate-based reservation is that LRD and self-similar behavior of traffic require a relatively high effective bandwidth to achieve a given end-to-end QoS (compared with Short Range and Poisson traffic). Reserving this effective bandwidth per-flow would decrease the network efficiency. To decrease the high traffic shaping delay at the edge nodes, partitioning of the end-to-end delay is used.

Partitioning (splitting) the end-to-end QoS [44] converts the end-to-end delay constraints into a set of local per-flow, per-link QoS guarantees [37]-[39]. Partitioning helps decrease the end-to-end delay by allowing better statistical multiplexing at each node. Each link in this case calculates and reserves the resources required for the satisfaction of its local QoS [37]. The partitioning of the end-to-end QoS can range from simple schemes such as assigning equal shares of the end-to-end QoS to links across the route such as [38] and [39] , to more sophisticated schemes that search for the optimal QoS partitioning [51]. The work in [38], [39], and [40] assumes an IntServ flow-based environment where each flow is allowed to have any QoS partition at any link by reserving the appropriate amount of resources. For this reason, these approaches are more concerned with partitioning the QoS requirement in a way that would minimize the resource usage. The partitioning problem is simpler in flow-based approaches because QoS assignment at each link is independent of other flows. These approaches do not fit a class-based QoS provisioning where there is no reservation and flows assigned to a given class experience the same QoS. In class-based approaches, the partitioning has to choose from the limited set of QoS levels provided at each link. The end-to-end QoS partitioning should guarantee the end-to-end QoS requirements of flows at the best possible use of the network resources. On the other hand, the choice of QoS partitions at each class would affect the routing decisions. Partitioning the end-to-end QoS for a route using a certain QoS class can not be done independent of the other routes. This combined choice of network wide route configuration, and QoS partitioning have received little research attention. The TE structure in this thesis aims to use this combined choice to proactively prevent QoS violations and hence increase the number of users. This raises the following questions:

- *Can a delay margin based TE include partitioning of end-to-end QoS?*
- *How to optimize the choice of the QoS levels maintained by the link QoS classes such as to ensure the satisfaction of all flow requirement while providing enough resources to accommodate future flow arrival ?*
- *How to include the WFQ weight or the queue priority in the TE formulation?*

1.3.4. Performance evaluation

The provisioning of per class QoS requires the ability to estimate each class performance for a given traffic input. In the rate-based reservation used in flow-based approaches, performance evaluation is only needed at the edge to calculate the effective BW required to achieve a certain delay. At edge nodes, it is easier to characterize the traffic because each flow represents a service. Using these characterizations, the effective bandwidth is calculated. The effective bandwidth is reserved in all subsequent nodes, hence minimizing the need for performance evaluation in these nodes. If end-to-end QoS partitioning is used, each link needs to estimate the expected QoS for a given amount of link resources in order to calculate resources needed to achieve the local QoS. To efficiently use the link BW, work-conserving schedulers are used. Work conserving schedulers introduce dependency between the queues. The dependency makes the service rate of one queue dependent on the traffic input and the bandwidth share of other queues. This dependency complicates the queue performance estimation. When flows are grouped in classes, each class input traffic is an aggregate of different traffic types. Such aggregates could display self-similar behavior. The self-similarity makes the performance evaluation more challenging. When self-similarity exits queues can remain occupied for long time periods, which magnifies the queue dependency problem. The long memory associated with self-similar traffic prevents simplifying the evaluation by employing Markovian models and approximations. Evaluating the performance of PQ and WFQ requires finding answers to the following questions:

- *How to characterize aggregates of self-similar traffic*
- *How can we analytically estimate the queuing performance of PQ and WFQ given self-similar traffic inputs?*
  - *How to decouple the effect of a queue on the other queues in way that would enable analyzing each queue as a FCFS queue?*
  - *How to evaluate the effect of changing the WFQ weight on the delay specially in the presence of self-similar traffic?*
- *How to include the performance evaluation in the overall traffic engineering structure?*

1.3.5. Centralized route configuration

The offline pre-computation can be performed at a central server, or it can be done in a distributed manner where each router pre-calculates and updates its own routing table. Accurate QoS routing requires a global knowledge of the network. Proposed state information advertising protocols such as OSPF provide techniques to collect this network state information. Although such protocols work in a distributed manner, the end-result is that each node has global information of the network topology and state. Each of these nodes can act in this case as a centralized routing server. This makes each node capable of calculating its own routes or calculating a routing configuration for the whole network. Letting each node calculate its own routes will render these routes optimal only from the point of view of this node (e.g. shortest paths are optimal from each path point view, but they do not provide the network with an optimal configuration). Letting one of the nodes act as a server and allowing it to calculate the routing configuration for the other nodes has many advantages. The processing power on this node can be increased to allow for more complex optimization algorithms. Since calculation will only be done at the server, the processing power of each node decreases as it is only used in the rare case of finding routes that are not pre-calculated by the centralized router. Using the global knowledge, and the increased computation power, the node can calculate a network wide optimal route configuration. Several centralized servers ([63]-[66]) have been proposed to handle different aspects of the network including route calculation.

The main concern with centralized servers is the reliability issue. The failure of a centralized server may hinder the network operation. Using a centralized server only for offline route selection, while keeping state advertising and packet forwarding distributed, decreases this risk. The distributed packet forwarding will depend on the routing tables calculated by the server. If the server goes down, the network continues to function and remains capable of establishing network flows. If the server remains down for a long period the network choices will be less optimal, but the network will remain operational.

1.3.6. Offline pre-computation

Routing involves three main tasks: state information advertising, route selection and packet forwarding. Route selection can be done online (as flows arrive to the network), or it can be offline. In offline route selection, a set of routes are pre-calculated based on the estimated traffic characteristics. The calculated routes are either used to update routing tables in all the network routers, or are stored at edge routers where source routing protocols such as RSVP and CR-LDP are used to establish the routes.

An arriving flow is directly forwarded or established over one of the pre-calculated routes without the need for online calculation. Although the offline calculation can be more complex and time consuming, it is only done once and routers do not need to calculate a new route for each arriving flow. If offline computation is done over a long time scale (every 2 to 6 hours) the total route computation (over time) in the network will decrease. Results in [45] show that, the use of pre-computation can reduce the cost of network QoS compared with on-demand path calculation. The concept of offline route selection has long been used. Using static link cost makes BE shortest path routing an offline route selection where routes are only calculated once and routing tables are used then to forward data.

1.3.7. Need for online approximation

Because steady-state QoS estimation is different [48] form the short-term QoS [46], it is expected that QoS violations can occur over the shorter time scales. The network traffic may deviate from that predicted by the traffic characterization. The optimization process complexity and the network stability concerns prevent rerunning the optimization more often to overcome short-term violations. It would be desirable to use the centralized offline traffic engineering over a long time scale. The long time scale allows for a more stable network and enables the use of complex optimization algorithms. Once the network configuration is determined, edge nodes will handle functionalities such as admitting flows in the network, establishing the flows and assigning flows to QoS classes. Core nodes will handle the forwarding of packets based on the configured routing tables.

To keep the network responsive to short-term variations, it would be desirable to use an online traffic engineering approach to handle route and QoS class assignment between successive long-term TE configurations. Both the centralized offline TE and the online short term TE should have the same objective and target optimization metric. This would ensure that routes obtained using the online TE would present the minimum deviation from the optimal network configuration calculated using the offline centralized TE. In this thesis we aim to use an online distributed TE that presents an approximation of the centralized offline TE approach. In order to do that, the thesis looks for an answer to the following questions:

- *What is the traffic engineering structure that would combine centralized offline TE, distributed online TE and their associated performance evaluation?*
- *How to combine delay margin based centralized offline TE with distributed on- line TE?*
- *How to allow nodes to adjust their link WFQ weights without explicit end-to-end resource reservation?*

1.4. Thesis Contribution

The main objective of this research is to develop efficient class-based end-to-end traffic engineering that maximizes the number of network users while satisfying their end-to-end QoS requirements. In order to achieve that, the thesis presents the following contributions:

- **Formulation:** Formulating the delay, class and route assignments as a delay margin based nonlinear optimization problem that reflects the inter-class and inter-link dependency introduced by the class based QoS provisioning approach and the end-to-end QoS requirements([49],[50]).
- **Solution:** Provide a problem solution in the form of three algorithms: The first two: centralized offline route configuration, and link-class delay assignment, operate in the convex areas of the feasible region to consecutively reduce the objective function using a per-link per-class decomposition of the objective function gradient. The third is a heuristic that promotes/demotes connections at different links in order to deal with concave areas that may be produced by a trunk route usage of more than one class on a given link ([49],[50])
- **Online Approximation:** Provide online approximations of the three algorithms to handle the routing of new connections in the periods between updates of the long-term centralized offline network optimizations [51].
- **Performance Evaluation:** provides an analytical technique to estimate the queue length and delay distributions for multi-queue systems using Generalized Processor Sharing (GPS) or Priority Queuing (PQ) strategies with time-correlated variable service rates, based on two-dimensional multi-level decoupling.([52]- [56])
- **QoS provisioning and Traffic Engineering Architecture:** The thesis proposes a delay-margin based Traffic Engineering (TE) architecture to provide end-to-end Quality of Service (QoS) in Multi-Protocol Label Switching (MPLS) networks using DiffServ at the link level.([58],[59])

1.5. Thesis Outlines

The next chapter provides the motivation for using a delay margin based traffic engineering approach. The chapter provides a detailed description of the QoS provisioning and traffic engineering approaches proposed in this thesis. The QoS provisioning and the TE structure are constructed such as to facilitate the implementation of the network optimization approach and the performance evaluation presented in Chapters 3 and 4. The main thesis contribution of delay margin based network optimization is presented in Chapter 3. Chapter 3 presents the formulation of the delay margin based TE optimization problem and provides the solution for the single class and multi-class networks. The chapter presents numerical results for networks using priority schedulers and weighted fair queue schedulers. Chapter 4 presents the proposed approach for analyzing the performance of priority queues and WFQs in the presence of self-similar traffic input. The approach in chapter 4 is used to analyze the queue length and delay survivor functions for priority queues and WFQs served with variable self-similar traffic. Chapter 5 presents the thesis conclusion and future work.

Chapter 2

Delay Margin based QoS Provisioning and Traffic Engineering

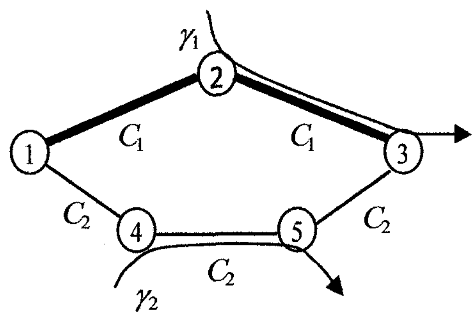
This thesis proposes the use of a QoS provisioning and traffic engineering approach that aims to maximize a network objective function based on the end-to-end delay margins of network connections. For each connection, the delay margin in this thesis defines the positive difference between the connection delay requirement and the end-to-end delay incurred by the flow in the network. The delay margin presents many advantages because it includes the connections end-to-end delays as well as their end-to-end delay requirements. Maximizing the delay margin requires minimizing the connection delay. To minimize the delay, the routing approach will choose shorter and less loaded links. Including the delay requirement in the objective function gives connections with lower delay requirements a higher priority. When choosing a route, the proposed approach will avoid adding a connection to a route that already supports connections with low end-to-end delay requirements. Adding traffic to a route increases the delay of links across that route. The increase in the link delays decreases the delay margin of connections that share the same link. Connections with smaller end-to-end delays are affected more by this decrease. To minimize the decrease in the delay margin objective function, it would be better to assign the new traffic to a routes that already has a high delay margin. By doing that, low delay margin routes are kept for use by connections that require small end-to-end delay. The delay margin also presents a method for mapping connections to DiffServ classes. Although adding a connection to the class presenting the lowest delay would present it with the best delay margin, it would also require the highest use of resources. This high resource usage will cause the biggest decrease in the delay margin of other connections. The use of delay margin based traffic engineering will push the connection to choose the DiffServ class that would present the connection with the lowest end-to-end delay, while presenting the least decrease in the delay margin of other connections. Another advantage of delay margin is that it provides an implicit idea about the resource reservations in the network. A route low delay margin implicitly means that there is not enough resources to support more connections. The low delay margin could be the result of many connections with high delay requirements or a few connections with low delay requirements. This means that by using a delay margin based routing the network does not need to explicitly reserve resources in the system.

This thesis proposes a traffic engineering approach that aims to balance the delay margin of different connections in the network. Connections with higher delay requirements that use longer route will have the same percentage delay margin as connections with small delay requirements that use shorter routes. Balancing the delay margin allows for diverting connections with higher delay requirements to longer routes. This leaves shorter routes to connections with lower delay requirements. To better explain the motivation for using a delay margin based TE and to initially describe the problem formulation, we use an illustrative example in the next section.

2.1. Illustrative example

2.1.1. Single class network

Consider an example network with the associated parameters shown in Figure 2-1. Assume that the trunk traffic in the example has packet inter-arrivals and packet lengths that are exponentially distributed and that packet service time are exponentially distributed and that delay is calculated using an M/M/1 queuing model [60]. Each end-to-end trunk in the figure generates an average traffic of γ and has a delay requirement of D. For an assumed single-class network, it is clear in the example that trunk 1, and trunk 2 will use link 2-3 and link 4-5, respectively. Trunk 3 will have two options: either to choose route 1-2-3 or 1-4-5-3. If shortest-path routing, min-delay routing or minimum interference routing (MIRA) [61] is used, route 1-2-3 will be selected. Although route 1-2-3 has the biggest capacity margin (lowest load) this capacity is unusable because the delay margin of trunk 1 which uses link 2-3 is small. Any load increase on link 2-3 will cause trunk 1 margin to go negative (causing QoS violation). Because of the possibility of a QoS violation, new flows choosing routes containing link 2-3 will be rejected. On the other hand, despite being longer and having links with less capacity margin, flows using links on route 1-4-5-3 have a bigger delay margin and hence can afford the decrease in delay margin associated with routing trunk 3 along this path. This means that the use of traditional load balancing, average network delay minimization, and network interference minimization will not help increase the number of network users. In other words, increasing the number of network users requires that the delay of each user be kept as far away from its requirement as possible in order to allow the network to accept more users without violating the delay requirement of the existing users. This simple example shows that even in a single network, there is a need for a traffic engineering approach that considers how far the delay of the routed calls is from their requirement (delay margin). This approach should aim at maintaining a positive delay margin for established connections.



Trunk	source	Target	γ	D
1	2	3	1	100
2	4	5	1	300
3	1	3	3	700

C1 = 12 BW units
C2 = 8 BW units

Figure 2-1: Example network

2.1.2. Multi-class network

When the capacity is shared between classes, the need to consider the end-to-end delay margin is further magnified. In a multi-class network, traffic engineering has to handle two more choices. It has to decide which class to assign to each flow and it has to decide how to partition the flow end-to-end QoS on links across the route. QoS partitioning becomes more difficult when a work conserving scheduler is used. Work conserving schedulers introduce coupling between classes on the same link. Assigning a lower delay partition (operating point) to a class would cause the delay experienced by other classes on the same link to increase. Consider again the example network in Figure 2-1 but assume that the network supports two delay classes and that the delay requirement of trunk 3 is tightened to 250ms. Because Trunk 3 endures a delay of 111 ms at link 1-2, and a delay of 200 ms at link at links 1-4 and 5-3, the end-to-end delay of 250 ms can only be supported on the 1-2-3 route. Supporting Trunk 3 on the 1-2-3 route would require link 2-3 to support a max delay of 150 ms. This delay cannot be supported if both trunk 1 and trunk 3 share the 2-3 link and would requires each trunk to use a different class. In this case, a big variety of class delay combinations could be achieved depending on the scheduling approach. Assuming the use of work conserving schedulers, then average delay of both classes is this example is related to each other using the following relation

$$\lambda \bar{d} = \lambda_1 \bar{d}_1 + \lambda_2 \bar{d}_2$$

where λ_i and $\bar{d}_i$ are the load and delay of each class, and $\bar{d}$ is the delay of FCFS queue fed with load λ where, $\lambda = \sum_{i=1}^{i=V} \lambda_i$.

Figure 2-2 shows the set of delay operating points that can be achieved for both trunks 1 and trunk 3 assuming that trunk 3 uses the 1-2-3 route. The figure shows that the delay requirement of trunk 1 could be satisfied with a range of delays varying from 90 msec to 100 msec. For this range trunk 3 end-to-end delay will vary from 248 msec to 242 msec. This raises the question of which delay operating point to assign to each class and the set of classes to assign to each route. Figure 2-2 shows an example choice of this point that would result in trunk 1 experiencing an end-to-end delay of 95 ms and trunk 2 experiencing an end to end delay of 245.

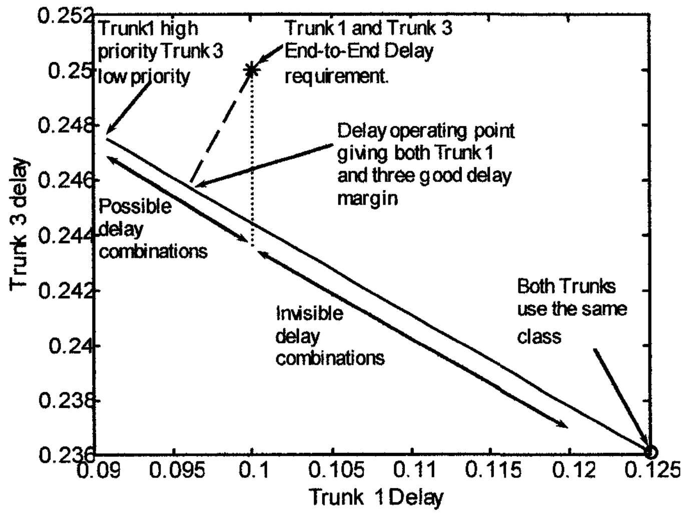


Figure 2-2: delay operating points

Min-hop routing is incapable of selecting the class, since both classes will have the same number of hops. Min-delay routing will always choose the higher-priority class to achieve the lowest delay and will not to assign the proper delay operating points to each class. Without per-class capacity partitioning, MIRA cannot choose a route. The ability to adjust the *delay operating points* provides more flexibility to accommodate a wider variety of QoS requirements. At a given class delay operation point, a flow assignment to a given delay (QoS) class decreases the *delay margin* of flows using other classes by an amount that depends on the delay operation points of these classes. This provides a trade-off in finding the appropriate flow to class assignment and delay operation point for each class. This simple example shows the need for a TE approach to deal with routing, class assignment, and the determination of the class delay operating point. This TE should consider the delay requirements of trunks to be routed, while preserving the QoS requirements of already established calls.

2.2. Proposed QoS Provisioning

This thesis aims at maintaining an end-user QoS that is similar to the one provided by flow-based QoS provisioning approaches. However, this thesis tries to overcome the scalability concerns of

IntServ by adopting techniques that do not require keeping per-flow state information at each node. The proposed QoS provisioning approach is based on providing end-to-end QoS by distributing the required delay on different nodes along the route. The thesis takes a different approach of trying to partition the end-to-end delay seen by a route instead of partitioning the end-to-end delay requirements of network flows. In other words, partitioning in the thesis is used to define the delay that each class will operate at. The delay operating point of a class at a given link is a function of the resources assigned to this class. In WFQ this resource assignment is governed by the queue weight assignment. Because queue weights are related to each other, increasing the weight of a given class means decreasing the shares of other classes in the link bandwidth. The queue relation introduced by resource sharing produces a wide variety of achievable QoS combinations (for a given set of queue input loads). This thesis uses an approach similar to [62] where IntServ edge networks provide accurate information about the end-user traffic characteristic but service in the network is not IntServ based. End-users in this case arrive individually at the edge of the network giving the proposed route configuration the control over the level of user aggregation and how much to load each route. The challenge in this case is how to maintain end-to-end QoS using this approach. The approach makes use of the MPLS route definition capability, but uses DiffServ schedulers at each node to provide per-link QoS. The core of the QoS provisioning approach is based on:

1) Maintaining a certain delay operating point at each class for each link.
2) Maximizing the LSP delay margin
3) Combining the choice of which DiffServ class to assign to an LSP with the choice of which route to use for forwarding the LSP traffic.
4) Allowing LSPs to switch their QoS class assignment at different links.

2.2.1. Class delay operating points

The delay operating point of each class is a function of the class input traffic, the input traffic to other classes, the link capacity, and the scheduler configuration. The delay experienced by an LSP is equal to the sum of the delay operating points of classes that the LSP goes through. By defining a delay operating point at each class the proposed QoS provisioning can guarantee an end-to-end QoS. The delay operating points can be adjusted such as to enable supporting the end-to-end QoS requirements. For an LSP with a low end-to-end delay requirements, the delay operating point of the DiffServ class used by the LSPs at each link can be lowered to ensure the satisfaction of the LSP QoS. The challenge is to adjust the delay operating points of all classes such as to ensure that the end-to-end delays of all LSPs are satisfied. The difficulty of doing that stems from the relation between different delay operating points on the network. Delay operating points are related on the same link and on the same route. The relation between the delay operating points of different classes on the same link is caused by the link resource sharing. Increasing the delay operating point of one class decreases the delay operating point for other classes on the same link. The delay operating points of the DiffServ class that the LSPs go through are related by the requirements that their sum should be below the LSP delay requirement. To maintain the same end-to-end delay, increasing the delay operating point on one link may require decreasing the delay operating point at another link. The ability to distribute the end-to-end delay through the adjustment of the delay operating points adds an important traffic-engineering feature. It allows for setting the delay operating points to high values at congested links and compensating for these choices by setting the delay operating points to low values at other links across the route. However, adjusting the delay operating points to ensure satisfying the end-to-end delay of all flows is more challenging in an MPLS-DiffServ network because several LSPs share the same DiffServ class at each link. A delay operating point that is good for one LSP may not be the best choice for other LSPs sharing the same class.

On the same link, adjusting the delay operating point can be done either by adjusting the scheduler parameters, or by adjusting the amount of traffic input to the class. The scheduler parameters can be adjusted such as to increase or decrease the resource share of a given queue. In a WFQ this can be done by adjusting the queue weights. The traffic input to the class can be adjusted through connection admission control, or by diverting a number of LSPs from one class to the other. For the same scheduler configuration, moving a number of flows from one class decreases the delay of this class, while adding them to another class increasing the delay on that class.

2.2.2. Delay margin

The QoS provisioning in this thesis aims to increase the end-to-end delay margin. The delay margin is the difference between the LSP requirement and the delay it experiences in the network. The delay that a flow experiences is equal to the sum of the delay operating points of classes that the LSP goes through. Keeping a delay margin has many advantages. The delay margin will prevent QoS violations in the case of instantaneous increase in the LSP traffic. An increase in the LSP traffic will cause an increase in the delay on classes that the LSP goes through. Adjusting class delays such as to maximize the end-to-end delay margins, will decrease the probability that a QoS violation will occur because of an instantaneous traffic increase. Keeping a high delay margin also allows for accepting more QoS connections. Accepting an LSP will increase the traffic input to the DiffServ classes assigned to the LSP. The traffic increase causes an increase in the class delay, which will affect all the LSPs going through this class (between different source destination pairs). If the delay margin of all LSPs sharing the same class can accommodate this increase, the new LSP can be accepted in the network.

2.2.3. Combine QoS mapping and routing

If nodes are able to categorize traffic into delay (i.e., QoS) categories rather than service type categories, it becomes possible to trace, to measure, and to satisfy end-to-end QoS levels Unlike traditional DiffServ, the QoS approach in this thesis does not separate mapping LSPs to DiffServ classes from routing. For each route, this approach presents a number of end-to-end delay options between each source destination pair. The end-to-end delay of a route is the sum of the delays presented by the classes along the route. In traditional DiffServ, this presents three delay options. In this approach, a flow assigned to class 2 on a shorter route can experience the same delay if was assigned to class 1 on a longer route. Assigning calls going through shorter routes to higher delay classes can alleviate the load on low delay classes, hence allowing these low delay classes to handle more users. In such schemes, an internal node will only admit new traffic if it is able to sustain the class delay category performance bound. This gives the network provisioning and management functions flexibility to govern the network in a way that approximately satisfies different end-to-end measures.

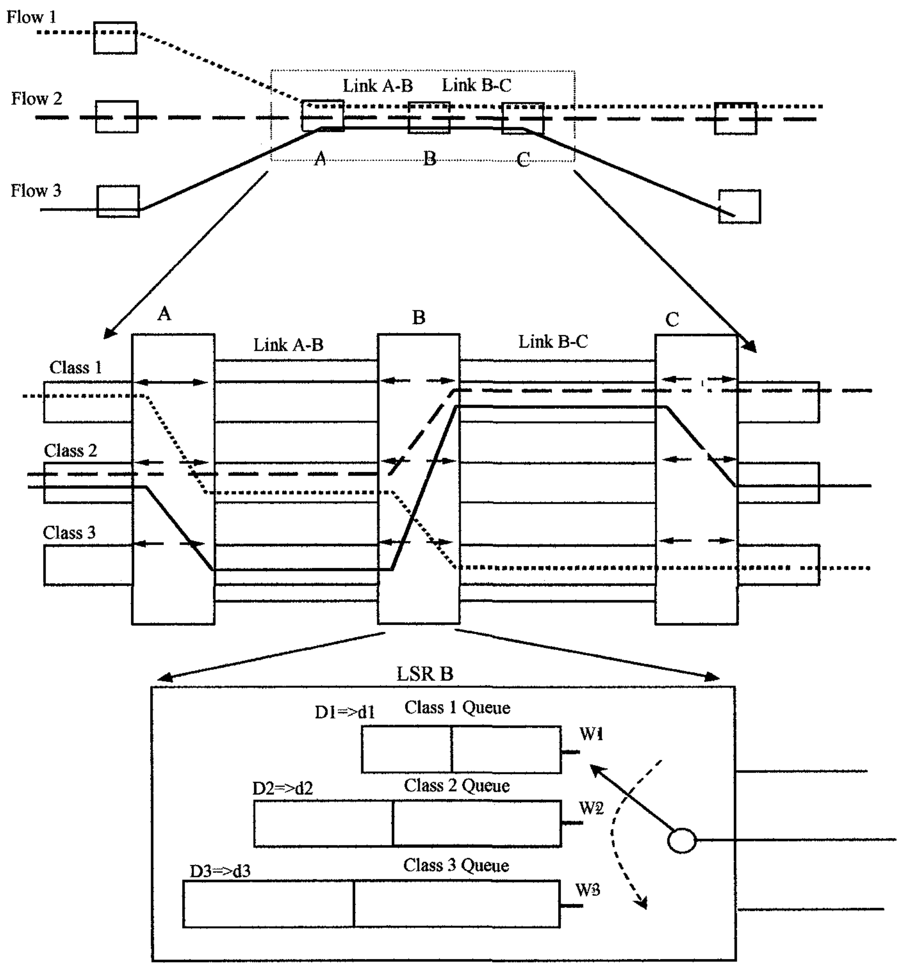


Figure 2-3: Example of operations in the proposed QoS provisioning

2.2.4. Switching classes

This thesis allows the same LSP to be mapped to different classes at different links. DiffServ assigns the same class to the connection on all links across the route. This limits the offered delay categories to the number of PHBs offered by DiffServ. DiffServ uses the same PHB behavior in all intermediate nodes, is mainly because the class assignment in DiffServ is done at the network edge. This class assignment is signaled to intermediate nodes through the DiffServ header. Allowing the route to use different classes at different links would greatly increase the granularity of end-to-end QoS. Consider an LSP route that goes through j links, where each link supports i QoS classes. Allowing different flows to switch classes along the route presents a number of QoS options equal to i^j, while using DiffServ only presents i QoS options. In the MPLS-DiffServ architecture, the PHB is signaled through the LSP header. Switching DiffServ classes in this case can be enabled by allowing the switching of the DiffServ PHBs with the switching of MPLS labels. Details and possible implementations of this variable QoS class assignment is described in [59].

2.2.5. Advantages of the proposed QoS provisioning

2.2.5.1. QoS provisioning Flexibility

The QoS provisioning represents an intermediate solution between IntServ and DiffServ QoS mechanisms. On one hand, it can operate as pure IntServ network by extensively increasing the number of delay categories such that each LSP has its own QoS class at each node. On the other hand, it can operate as a conventional DiffServ MPLS network by assigning categories at the edges without switching delay categories at internal nodes.

2.2.5.2. Wider spectrum of QoS guarantees

The QoS provisioning renders a possible larger pool of end-to-end delay characteristics (edge-to-edge delay classes) for the same number of delay categories as a result of possible category switching at internal nodes.

2.2.5.3. Better network utilization

With the availability of a wider range of QoS guarantees, potential better network utilization can be anticipated as a result of higher probability of satisfying relatively accurate QoS demands of all network LSP.

2.2.5.4. Maintains simple forwarding

Delay categories assignment is only performed during LSP path establishment. Consequent traffic forwarding remains simple with the same label switching mechanism.

2.2.5.5. Surviving with bottle-necks

The QoS provisioning provides a means for network surviving with bottlenecks. An LSP may afford the bottleneck consequent delay as long as it is assigned better delay categories at other nodes of the path.

2.3. Delay Margin Based Traffic Engineering

The inclusion of end-to-end QoS in the TE complicates the problem of finding the set of routes that can transport all the end-to-end flows between network edges. It adds the problem of guaranteeing the flow QoS satisfaction to the route configuration problem. The proposed TE in this thesis focuses on combining QoS provisioning and routing in order to provide end-to-end QoS using a class-based QoS provisioning. The objective is to map end-to-end flows on routes, assign them to QoS classes, and partition the end-to-end QoS that they experience into per-link QoS operating points. This section provides a description of the TE approach proposed in this thesis. The traffic engineering acts as a platform for applying the algorithms provided in chapter 3 . It also makes use of the performance estimation techniques that are provided in chapter 4. The remaining of this chapter provides detail about this proposed TE.

2.3.1. The proposed TE process and context

This thesis presents a traffic engineering approach that is based on the use of delay margin based objective function. TE is an iterative process that has to continuously evaluate problems in the

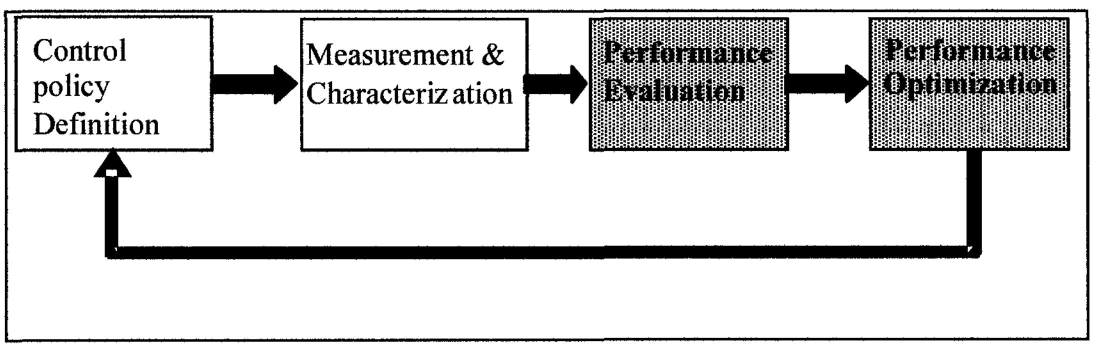


Figure 2-4: Traffic engineering cycle

network in order to change the network configuration, control methods, and network tuning parameters. The TE cycle shown in Figure 2-4 was proposed in [21].

In general, traffic engineering a network goes through four phases, the first phase defines a set of control policies based on a set of given network objective functions, network utility models, optimization criteria and constraints. In the second phase, the current and future state of the network is characterized using either/both online measurement, or offline prediction. In the third, analytical or simulation approaches are used to evaluate the current network state. Performance evaluation of the network is used to: (1) analyze the current network performance, (2) identify current and future problems, (3) evaluate the performance of various problem resolution approaches and, (4) find the optimal network configuration. The network configuration is represented by: route definitions, resource allocation, queue configuration, traffic mapping, and traffic control. The optimization process will result in changing the network operating conditions, and will affect the network control policies and hence would require repeating the cycle.

The proposed delay margin based traffic engineering covers all stages, however the thesis research contributions are mainly focused on the performance optimization and performance evaluation. The thesis focuses on the performance optimization and uses control policies that represent distributed approximations of these performance optimization approaches as will be presented later. Both the optimization and the control try to introduce minimum changes to existing QoS provisioning approaches and protocol.

2.3.2. Proposed traffic engineering context

The context of Internet TE [21] pertains to the scenarios where TE is used. It can be a network context, a problem context, a solution context or an operation and implementation context. An outline of the issues associated with each context is shown in Figure 2-5. The work in this thesis will focus on the problem context and solution context of traffic engineering. In the problem context, the thesis aims to provide a formulation of the problem that considers end-to-end QoS, as well as inter-class QoS dependencies that are introduced by the multi-class queuing discipline.

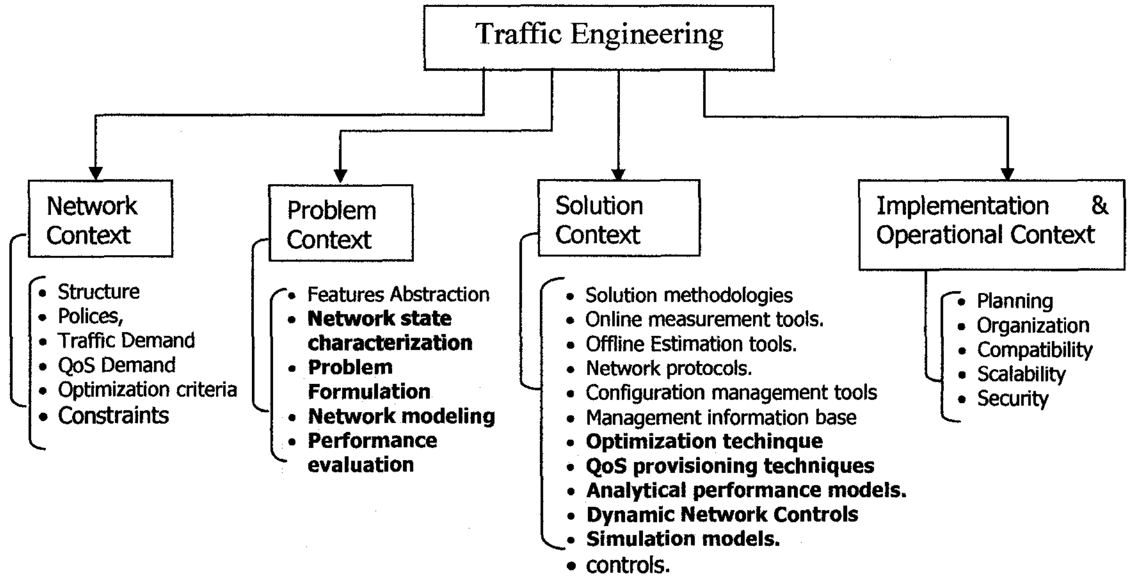


Figure 2-5: Context of TE

The formulation requires the modeling of end users, their flow aggregates, the network links, and the scheduling used at each link. Performance evaluation models are required to enable finding a solution for the optimization problem. In the solution model, the thesis tackles research issues related to optimization techniques and approaches needed to find an optimal or near optimal solution of the route and QoS assignment problem. The thesis also deals with the non-separable problem of QoS provisioning both from an end-to-end flow perspective and from a local per-link perspective. The solution context related to QoS provisioning includes a static part related to deriving analytical model for the estimation of the delay given certain link resources, specific long-term input traffic characteristics, and for a given scheduling configuration (e.g. weight or priorities). The Dynamic part will have to deal with distributed online control to adopt to short-term traffic variations. The online short-term QoS control is difficult to model analytically. Simulation models are used to analyze these online techniques. The thesis research assumes that the network context issues such as structure, traffic demand, and QoS requirements are pre-defined and will use them as input to both the simulation and optimization techniques. Although the thesis research is intended for an IP implementation context, the approaches can be used for any class-based QoS provisioning. The proposed approaches are made general (less specific to MPLS and DiffServ) in order to make the results usable within different implementation contexts.

Table 2-1 Taxonomy of the proposed TE

TE level	Topological level	Time Scale level	Centralized vs distributed	Offline versus online
1	Network	Long-term	Centralized	off-line
2	Flow	Medium Term	Distributed	Online
3	Link/Class	Short Term	Distributed	Online

2.3.3. Proposed TE structure

The traffic engineering proposed in this thesis works on three time scales. Each time scale operates over a different topology level. Table 2-1 shows that each time-scale is associated with control over a topology scale. The time scales correspond to the speed of change on their associated topologies. The longer time scale (hours to days) is associated with network-wide TE. A medium time scale that operates on a range of minutes to hours deals with the LSP-level TE. The short time scale TE operates on a range of seconds to minutes, and deals with queue configuration and scheduling issues. The time scale and topology combination allows each level to efficiently manage the network resources. Network wide computations are performed less frequently because they are more complex and require more network state collection. On the otherhand operations that require faster response such as maintenance of per-link QoS and performance estimation of per-link per class QoS are continuously done (every shorter time frame). This thesis adds an intermediate time frame where online approximations of the centralized approaches are used to try to achieve the best possible network configuration between the more accurate but less frequent offline calculation of the network configuration. Other options are associated with each level such as whether the approach is centralized or distributed and if the decisions/calculations associated with each level are performed offline or online. A summary of the proposed TE taxonomy is shown in Table 2-1 and the TE block diagram is shown in Figure 2-6. More details about each level are presented in sections 2.3.4, 2.3.5 and 2.3.6

2.3.4. Long-Term network TE:

On the long-term TE, routes and their QoS class assignments are offline computed using the centralized offline route configuration presented in chapter 3. The optimization process is based on long-term multi-scale traffic characterization of the input traffic. The approach assume the presence of a centralized server to make the calculation and distribute the resulting configuration on network routers.

The long-term TE in this thesis assumes that during a small time interval, the change in the long-term stochastic characterization of the traffic is small. The slow change in the traffic characteristics stems from the slow change in the behavior of end users that generate this traffic (e.g. an end user usually browses the web for several minutes). The slow change in the long-term traffic behavior enables characterizing this behavior with reasonable accuracy. The long-term traffic engineering approach proposed in this thesis uses the long-term traffic characteristics to configure the LSP routes and to assign DiffServ class to these LSPs at different network links. The route and class assignment in this proposed TE aim at maximizing the overall end-to-end delay margin. Maximizing the delay margin enables the network to handle future variation in traffic pattern.

The long-term TE in this thesis uses a centralized server for the calculation of the network route and class assignment configuration. The centralized server enables collecting global network state and can have the processing power to overcome the complexity of the route and class configuration. Once the optimal route configuration is calculated, the routing tables corresponding to these configurations are downloaded to the network routers and are used for forwarding the flows. The centralized server operates on the traffic averages over time period that range from 30 minutes to several hours. The change in this averages traffic variation is the result of either the traffic self-similarity or long-term user behaviors. By having a global knowledge of the network, the long-term TE adjusts the network so that in average the network will have a high delay margin. It is then the responsibility of the medium term flow level TE and the short term link level TE to handle the variation around these long term averages. The existence of the high delay margin provides a better operating conditions for the medium and short term TE and allows them to accommodate more traffic flows. The long term TE also works to correct traffic configuration that result from the medium term and short-term TE lack of global network state. By reconfiguring the network, the long term TE re-adjusts the delay margins cost associated with each link and hence biases the route choices taken by the medium term. The network will continue to use this routing configuration until a new TE update, and hence can operate for hours without the centralized TE server. The frequency of repeating the long term TE depends on the nature of the network traffic. The repetition can be event driven or time driven. In the event driven approach, the long term TE is triggered when the delay margin falls below a given threshold. In time driven approach, simulation or analytical models based on the long-term network traffic characterization can be used to find the best update inerval interval.

The use of centralized routers is quite common in network traffic engineering ([63],[64],[65],[66]). Bandwidth Brokers [67] that are usually associated with class-based QoS provisioning [68] can be seen as centralized servers that manage the DiffServ SLA. Simulations in [70] confirm that using centralized servers deliver higher traffic acceptance rate and lower flow blocking rate when compared with distributed solutions. Results in [45] show that when paths are pre-computed every large update interval (150 sec to 300 sec) they can reduce the per-node processing time computation but with a proportional loss in the bandwidth acceptance ratio compared with on demand path computation. The difference between these server approaches is mainly in the role they play in the TE process and the optimization technique they apply. While RATES is based on minimum interference algorithm [61], TRES is based on maximizing the minimum remaining bandwidth [71]. A similar approach to this thesis research proposal is WISE [64] where a centralized offline TE Server provides offline calculation of routes and allows for distributed short-term correction.

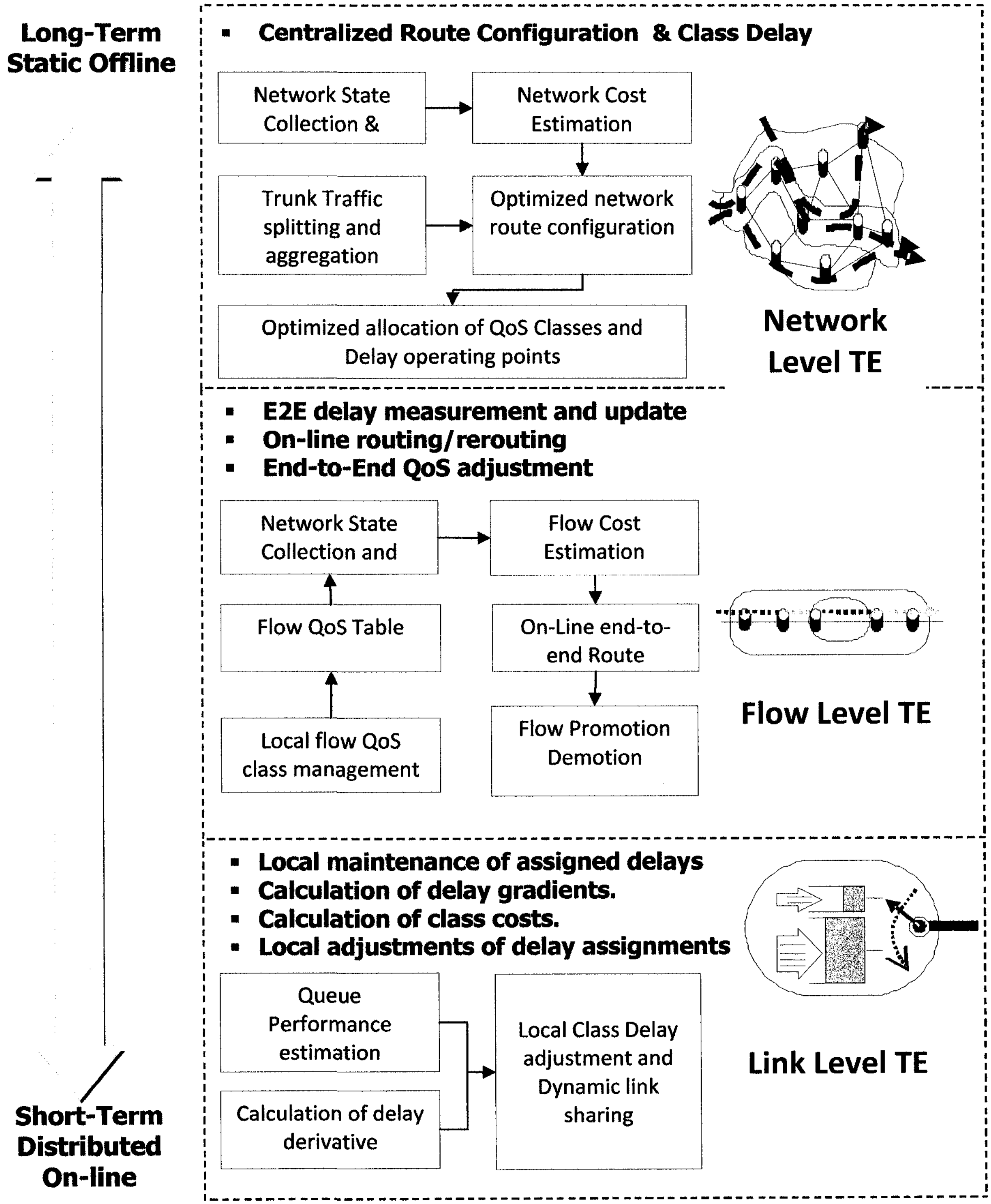


Figure 2-6: Hierarchy and functions of the proposed TE approach

Unlike RATES [63] (Routing And TE Server) and TRES [66] which assume an IntServ environment, the proposed server in this thesis operates on a DiffServ environment. IntServ approaches assume isolation between flows and hence can control the resource reservation for each flow. Traffic engineering in this case reduces to only configuring the network routes. None of these servers was intended for the thesis objective of a class-based end-to-end QoS provisioning. In a class-based end-to-end QoS, there is no resource reservation and end-to-end QoS is a function of the class delays that a given route goes through. On the other hand, assigning a flow to a given class at a given node causes this node class delay to increase and hence, the end-to-end delays of all flows using this class. To properly traffic engineer the network, the server in this thesis has to handle routing, class assignment as well as find the appropriate delay operating point for each class. The optimization problem in the thesis has increase the number network users by implicitly considering the users end-to-end QoS requirements to maximize the average unreserved BW.

Chapter 3 will provide a detailed definition of the formulation of the problem and the centralized approach used to find the routing, class assignment and delay operating points. The related work shows that the concept of centralized server is valid, and has been used by a number of TE approaches. However none of these TE approaches considered both route and class assignment in a DiffServ environment. Although there is also an open research issue regarding the structure and functionality of the TE server, this thesis only assumes the presence of this server functionality and focuses on the optimization problem associated with that server.

2.3.5. Medium-term flow level TE

The central offline network configuration is calculated every long time interval. To take care of flows that arrive in the network between two long-term network configurations this thesis uses a medium term flow level TE. The flow level TE handles the actual forwarding, establishment, and resource assignment at the edges of the network. The flow-level TE in this thesis operates at the network edges to deal with functions such as accepting end-user connections inside the flow and estimating the statistical characteristics of the flow traffic from it constituent end-user statistical characteristics. To keep the network operating close to the optimal point calculated using the long-term TE, the flow level TE monitors each route end-to-end traffic and its received QoS and reacts in order to calculate the delay margin. Monitoring the delay margin of each flow allows for implicitly monitoring the reservation status on links that the flow goes through. The flow-level TE aims to make online routing and class assignment that present the least deviation from the configuration obtained using the centralized offline traffic engineering. The objective is to make the network more stable by minimizing the need for offline configurations. The flow level TE can do that either by diverting the traffic to another route or by changing the DiffServ class assignments across the route. By demoting routes at congested links to DiffServ classes with higher delay and promoting them to lower delay class at consecutive links. The flow promotion demotion can release resources on congested links that can be used to accommodate more connections.

The operation of the flow-level TE requires cooperation at the link level. The flow-level TE needs to know the cost of assigning a given QoS class to the flow at a given link. It also requires the ability of the link to change its scheduling parameters and queuing configuration to adjust the delay operating points. The flow level TE assumes the use of an RSVP like protocol for the establishment, maintenance, and release of the flow routes. For each established flow, the TE keeps a record of its QoS requirements in the *Flow QoS table* at the edge of the network. Although per-flow information is kept at the edge of the network, this information is not kept at the link level. In line with the DiffServ concepts, each class on a link is rather required to keep an aggregate state information about all flows going through it. This aggregate state is a function of the End-to-End delay margins of all flows using the link QoS class. As shown in Figure 2-6 the *Flow State Collection and update* operates at the edge of the network and measures the QoS received.

The flow end-to-end QoS is measured using either piggyback end-to-end network monitoring messages, or is collected using state collection messages such as PATH message in RSVP. The measured delay is used to calculate the delay margin and the same set of messages are used to report this margin to all the links used by the flow. It is to be noted that a flow end-to-end state collection is not limited to its QoS classes, but that it can collect QoS measures about other QoS classes across its route. The *Flow Cost Estimation* module estimates the expected cost increase/decrease associated with promoting/demoting a flow using the collected data.

The online approximation of the route configuration evaluates the possible route choices for an arriving flow and chooses a route using link costs from the *Flow Cost Estimation* module. The *flow promotion/demotion* module handles the actual promotion demotion as described in [59]. An approach similar to the promotion demotion approach used in this thesis was proposed in [72]. It changes the end-to-end DiffServ class assignment to maintain end-to-end guarantee. The QoS options in [72] are limited since it uses the fixed class assignment. Another DiffServ approach [73] maintains end-to-end QoS by allowing the promotion and demotion of DiffServ marking in links across the route. Changing the marking at certain nodes is more efficient than changing the end-to-end QoS PHB as in [72]. The problem with the approach in [73] is that it changes the precedence of packet loss and it limits the marking of DiffServ to 2 instead of three. The closest to the thesis proposed approach is [74] where packets are allowed to choose between a set of classes at each node. The network configuration in [74] is based on a stratified best effort network and hence there is no delay limit associated with each class. A similar concept in [69] partitions the flow end-to-end delay into per link delay budgets. At any given router, the approach in [69] increases the flow priority if it experiences delay in excess of its total assigned delay budgets up to this router [69]. The approach in [69] increases the flow priority at a given router. However, the approach in [69] cannot fit this thesis objective as it requires the definition of per-flow per-link delay budgeting which cannot be achieved in DiffServ environment.

2.3.6. Short-term link level TE

The short-term TE has three main roles. The first role is to locally estimate delay and calculate the local link cost. The second role is to adjust the scheduling parameters in such a way that enables achieving the QoS operating points. The third role is to adjust the delay operating points such as to increase the overall delay margins of flows using the network. The link-level TE will do that through the proper assignment of queue configurations (queue length, queue priority or weight). Several techniques have been proposed for the dynamic link resource sharing in class-based QoS systems. WFQ inter-class delay assignments were mainly concerned with adjusting the weights given a local delay requirement for each class. Dynamic control techniques were used to adapt WFQ weights in response to measured input traffic patterns [75], average queue [76], queue length [77] the moving average input rates [78], or the delays of served packets [79]. Most of these techniques assume the existence of some per queue QoS requirement and work to maintain the queue performance close to this value, or to maintain the QoS received by all queues proportional to their QoS requirements. Although these weight adjustment techniques determine the QoS received by each queue, they are not concerned with defining the QoS operating point of each queue. Because the determination of the delay operating points requires the existence of performance evaluation methods, the thesis assumes that these methods will be used for the initial determination of the queue weights and that dynamic techniques such as ([75] - [79]) will be used for the fine tuning of the queue weight configuration. The inter-class delay operating point assignment in this thesis is more concerned with choosing the delays of different classes sharing the same link in such a way that will maximize the margins of routes assigned to these classes. It considers the interdependency between *classes* (due to WFQ) and between *links* used by network routes. Unlike other techniques, the determination of these delay operating points in this thesis based on the same network wide objective function that is used by the routing and class assignment approaches.

The objective in this thesis is different in that it aims to increase the end-to-end delay margins. Because end-to-end flows use several links, the TE should be such to allow the delay operating point adjustments on different links to work together in order to achieve the same objective. The block diagram for the link model used in the thesis in shown in Figure 2-7. As shown in the figure, each link monitors traffic on all of its classes. To investigate the effect of adding a flow to a given class and, the characteristics of the arriving flow are used to analytically estimate the characteristics of the aggregate using the approach that will be detailed in Chapter 4. The multi-class performance analysis introduced in Chapter 4 is then used to estimate the QoS of each class. Results from the performance analyses is used both for connection admission and for adjusting the WFQ queue weights based on the delay operating points calculated by the centralized schemes that will be presented in Chapter 3. Results of both connection admission and weight adjustment are fed back to the performance estimation block to be used in subsequent evaluations.

2.4. QoS Performance Estimation

Analytical performance estimation is critical to the operation of the three TE levels. On the network level it is used to estimate the actual delay received at each class as a result of routing a given flow through this class. Knowledge of the delay received at each class is used to guarantee the flow end-to-end delay. The cost function used in the centralized offline optimization process is a function of the traffic input and delay received by each class on each link. Being able to analytically derive the change of this cost with the change of network configuration is vital to the optimization process. On the flow level, it is used at each link to define the effect of promoting

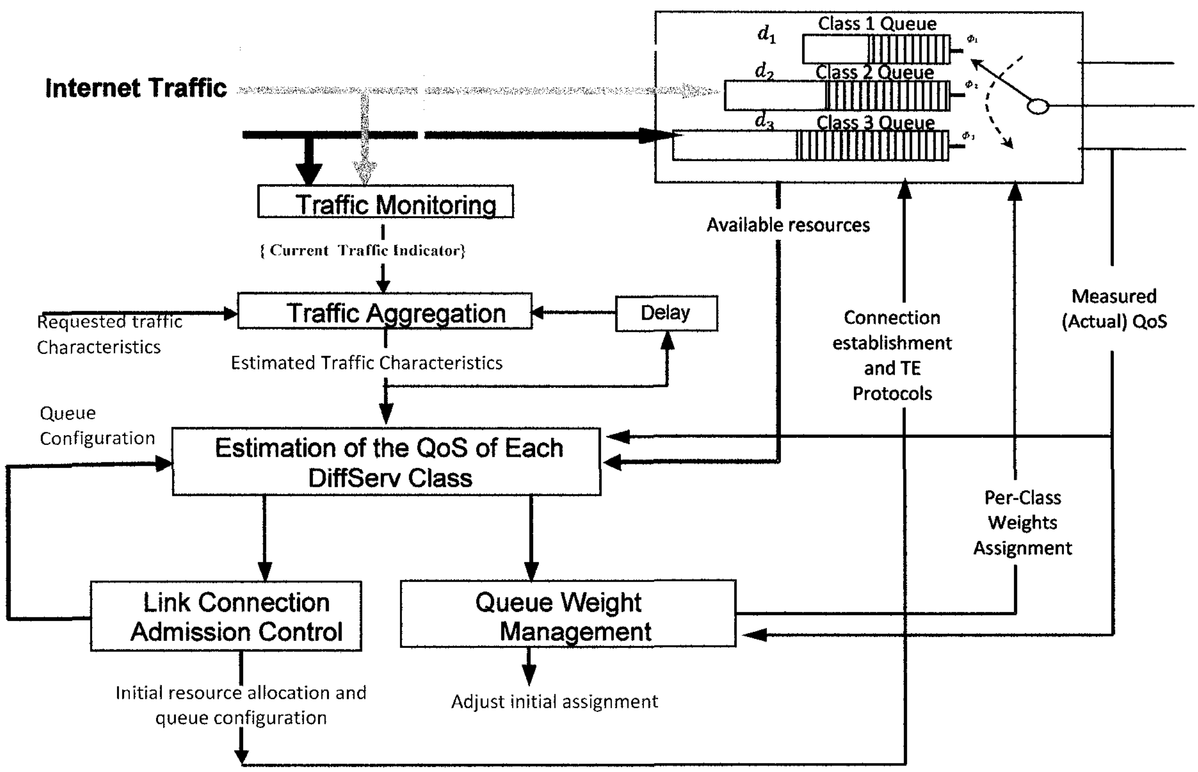


Figure 2-7: Link Model

or demoting a flow on the class received delay, and to find the links where the promotion will cause the least increase in the network objective function.

The performance evaluation in this thesis assumes the use of multi-scale traffic models at the measurement and characterization phase and focuses on the analytical evaluation of multi-class queue performance. Multi-scale traffic modeling was chosen because of it ability to model Long-Range Dependent traffic as well as a wide variety of traffic types. Both performance evaluation and optimization are closely related to each other. The existence of analytical performance models enables the optimization to accurately search for optimal network configuration.

Chapter 3

End-to-End Delay-Margin Based Optimization and Algorithms

3.1. Introduction

In Chapter 2, we initially introduced the importance of considering the delay margin in the network traffic engineering. Chapter 2 also presented the traffic engineering structure associated with delay margin TE. At the core of this TE approach is the problem of delay margin based route configuration, class assignment, and the definition of each class delay operating points. This thesis chapter formulates a nonlinear optimization problem that combines the delay, class and route assignments problems. The formulation is based on a multi-class delay margin penalty function. This delay margin penalty function approaches infinity as the delay margin of any end-to-end connection approaches zero. For a single-class network the function is convex, hence allowing for convex optimization techniques. In the multi-class case, this chapter shows that concave areas appear in the objective function. These concave areas are the result of the inter-class dependencies. The chapter presents a solution of this nonlinear optimization problem that is based on three algorithms. The first two algorithms, *centralized offline route configuration* and *link-class delay assignment*, operate in the *convex* areas of the feasible region to iteratively reduce the objective function using a gradient-based approach. Both solutions are based on a decomposition of the function gradient into per-link, per-class components. Each component captures the effect of all flows using this class and the interacting flows. The third algorithm is a heuristic used at the LSP level to promote/demote LSP at different links in order to move across the *concave* areas. Based on the nonlinear optimization problem, we present approximations that are suitable for *distributed online* operations. Both the *offline* and *online* algorithms are intended for use in the TE structure presented in Chapter 2. Although the TE proposed in Chapter 2 is general enough for use with other approaches, it was refined to enable implementing the algorithms presented in this chapter. The refinements aims to integrates the functionally of different algorithms and provides the platform for their operation.

This thesis chapter starts by providing the related work in terms of problem formulation and solution. It then provides a detailed formulation of the problem. The formulation assumes the use of work conserving schedulers such as the weighted fair queue and explains the source of the concave areas in the nonlinear optimization problem. In Sections 3.4 and 3.5 the offline and online algorithms are presented. Illustrative results are shown in sections 3.6, 3.7, and 3.8 for a single class network, a multi-class network based on weight fair queue schedulers and a multi-class network based on priority schedulers

3.2. Related Work

Techniques that handled QoS in routing either optimized the choice of a route for a single flow or they consider optimizing the route choices for the whole network. Flow-by-flow routing focused on optimizing the choice of single routes. It used well known network flow optimization problems such as the shortest path, the maximum flow or minimum cost flow [80]. Flow by flow QoS routing either included the flow QoS in the objective function or used constrained versions of basic flow optimization problems. A more complete survey on constrained QoS routing approaches can be found in [16]. These approaches do not fit our objective as they either use flow-by-flow (per-flow) reservation, or assume traffic-invariant network QoS. The second approach considers an objective function that represents all flows in the network. The optimization in this case presents a variation of the multi-commodity flow optimization problem [30]. Basic multi-commodity flow optimization included QoS in the objective function in order to optimize the network over all QoS. Maximizing an over all QoS function does not fit the objective of the TE proposed in this thesis, as it does not guarantee the QoS satisfaction of all flows in the network. On the other hand, considering end-to-end QoS (e.g delay) as constraint makes the problem more complex.

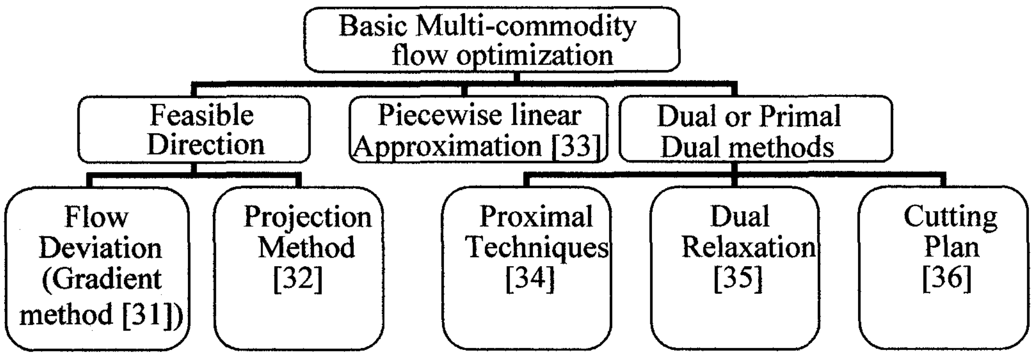


Figure 3-1: Multi-Commodity flow optimization techniques

The centralized approach in this thesis can be seen as multi-class constrained version of the basic multi-commodity flow optimization problem [30]. The solution of any multi-commodity flow optimization problem uses 3 main steps. The first step decomposes the main optimization problem into a set of simpler sub-problems (possibly linear optimization problems). This can be done by decomposing the cost function into per-link cost functions or end-to-end cost functions hence transforming the network optimization problem to sub-problems of link or flow optimizations. In the second step, optimization of these sub problems is conditioned on the existing solution of the main problem (for each sub-problem the other sub-problems look as constant). Each sub-problem is then solved in the third step in a way that would minimize the over all objective function and a new solution is found for the main problem. These three steps are iteratively repeated until a given stopping condition.

Several techniques have been proposed for the solution of multi-commodity flow optimization [30]-[36] as shown on Figure 3-1. These techniques focused on enhancing the basic multi-commodity flow optimization algorithms by varying the decomposition technique, the cost functions, and the sub-optimization problem but did not consider the end-to-end delay, or the capacity issue and they were mostly based on average load. A brief comparison between them is provided in Table 3-1. A detailed survey of these techniques can be found in [33]. The optimization approach used in this thesis starts by using a centralized offline approach that consider all network routes at once and then uses it to provide an approximate distributed approach that can be used on a flow-by-flow basis. To do that, the thesis uses a gradient based approach for routing that is similar in concept to flow deviation [31]. It is shown later in this chapter that the dependence on the gradient provide an online distributed approximation to the centralized technique. Although the function Hessian can be calculated it is very difficult to decompose and hence it is difficult to provide an online approximation.

Table 3-1 Comparison between Multi-Commodity flow optimization techniques

	Flow Deviation (gradient method) [31]	**Flow Deviation (Projection method) [32]**	**Proximal Techniques [34]**	**Dual relaxation [35]**	**Cutting Plan [36]**
Decomposition	by link	by route	distributed on route and link	distributed on route and link	distributed on route and link
Sub-problem	Shortest Path	Shortest Path	Non-linear minimum cost flow problem	Descent direction	Shortest path
Number of iteration	Big	Low	Medium		Very low
Iteration complexity	Low	Very Low			High

The basic multi-commodity flow optimization techniques [30]-[36] cannot be directly used within the framework of the TE proposed in Chapter 2 because of the differences imposed by the TE objectives and structure. A comparison between the optimization problem considered in this thesis and the basic multi-commodity optimization is seen in Table 3-2.

Table 3-2 Difference between Basic and Multi-class Delay Constrained Multi-commodity Problem

	Basic Multi-commodity Problem	**Multi-class Delay constrained Multi-commodity Problem**
Combination of route and Delay Class	Routes are defined by the set of links, and no delay classes are defined on the link	Routes are defined by their links and their QoS class assignments and the optimization algorithm looks for the set of links making the route as well as the Delay classes assigned to them(which affects end-to-end delay)
end-to-end Delay requirement	No end-to-end constraint	The route and delay class assignment should be such as an end-to-end delay constraint is satisfied
Traffic Characterization	Trunks and links traffic are characterized by their average traffic	Trunks are characterized by their average traffic and end-to-end delay requirement while links are characterized by their average traffic and their class delay operating points.
Capacity partitioning	The capacity is shared among all Trunks	Class delay operating points determine the number of flows assigned to each class and weight assigned to each class
Cost function	Single dimension	Multi-Dimension

As shown in the table the main differences can be summarized in three main points. The first point is the existence of End-to-End delay constraints. The second is the need to deal with the combined problem of routing and delay operating point definition. These two problems are not limited to multi-class networks, but they are applicable in single class networks as well. The third problem is imposed by the existence of multi-classes. The link multi-class work conserving schedulers introduces coupling between classes and adds the problem of end-to-end class assignment to the routing and delay operating point assignment problems. .

To deal with the delay constraint this thesis uses an end-to-end delay margin barrier function for each flow. As the delay of a given flow approaches its requirement the function goes to infinity. The same barrier function is also used as the objective function in order to maximize the overall network delay margin and hence enhance the over all network delay satisfaction and increase the number of QoS users in the network (as will be shown later). The use of barrier functions [81] is a well known technique to eliminate inequality constraints in nonlinear optimization.

To deal with the combined problem of routing, delay operating point definition and class assignment (when multi-classes are considered), the decomposition approach used in this thesis acts on two levels. In the first level, it decomposes the problem into three sub problems: a routing problem, a per-link definition of delay operating points, and a class assignment problem. In the second level, each of these problems is partially decomposed on a per-link per class basis. The thesis will show later that full decomposition is not achievable because of the existence of the per-flow end-to-end delay requirements. The three problems are then repeatedly solved one after the other until a stopping condition is reached. The idea of using the same cost function and decomposing the problem into two sub-problems was used to combine flow assignment with capacity allocation in [83] and [82] by repeatedly solving the two problems one after the other. The approach in [83] used two objective functions one for capacity and one for delay and optimized each conditioned on the other. On the other-hand, the approach in [82] used a function that is the sum of both components but split the optimization into two problems. The first problem provides a Lagrangean relaxation of the problem and the second uses a sub-gradient optimization procedure to generate feasible solutions by improving the quality of the Lagrangean lower bound. Consecutive iterations between flow deviation and the solution of linear equations is used in [84] to get the route and capacity allocation that maximizes a network revenue objective function.

In this thesis the capacity assignment is replaced by the definition of per-link delay operating points. The idea to split end-to-end QoS delays into a set of link QoSs, and let each of these links locally monitor and guarantee its local QoS bound was introduced in [38]. Several techniques dealt with the partitioning of different QoS aspects. A partitioning of the loss rate guarantees while optimizing bottleneck utility using a heuristic was proposed in [38]. The approach uses a heuristic that is based on either an even or proportional partitioning of QoS between links. It then optimized resources by recursively choosing the tightest QoS in a link and redistributing end-to-end QoS bounds. Similar to [38], the work in [39], divides the end-to-end delay into pieces and uses a greedy algorithm to gradually distribute the total delay piece by piece, giving each piece to the link where it most improves the probability of success. The QoS partitioning problem is solved in [89] using greedy add and greedy move algorithms that are based on proves in [90] and [91]. Both algorithms in [89] iteratively move a small amount of link resource from the highest-cost gradient link to the lowest one. The distributed version of the delay assignment algorithm in this thesis uses a similar concept of incremental adjustment, but it operates in a DiffServ environment. The work in [38], [39], and [40] assumes an IntServ flow-based environment where each flow is allowed to have any QoS partition at any link by reserving the appropriate amount of resources. For this reason, these approaches are more concerned with partitioning the QoS requirement in a way that would minimize the resource usage. These approaches do not fit the class-based QoS provisioning used in this thesis, where there is no reservation and flows assigned to a given class are of the same QoS. The problem of combining routing and QoS partitioning investigated in [82] uses an IntServ model and the assumption of unicast links. A combined approach based on inaccurate traffic information was introduced in [85]. Although the solutions in [39], [85] and [82] take a similar approach to the one used in this thesis by combining routing and QoS partitioning, they are flow-by-flow approaches that operate in an IntServ environment. An algorithm for network-wide QoS partitioning of flows sharing the same class/link was introduced in [86]. Unlike [39], which only focused on maximizing the delay partition at each link, [86] uses the end-to-end delay inter-link dependency to allow some links to decrease their delay to provide slack for other links.

The approaches discussed above are either based on single-class networks or they use the assumption that each link capacity is partitioned between classes. This partitioning allows the network to be modeled as number of single-class networks, each for one class (e.g [63] and [87]). Similar to [86]. the approach in this thesis considers delay partitioning in the whole network and hence has to consider the interlink dependency. However unlike all of the approaches above this thesis is not interested in partitioning the end-to-end delay requirement but is rather interested in choosing the lowest possible delay operating point for each link such that the end-to-end delay margin is maximized. This problem is quite simple in single class network, however when multiple queues are used at each link, the inter queue dependency complicates the problem. Multiple queues are used to increase resource efficiency, but they introduce two problems. The first problem is the QoS interdependency between different classes and hence, the network cannot be modeled as a number of separate networks. The second problem arises from the capability to operate with a wide variety of QoS combinations for a given set of queue input loads.

The use of a network wide objectives function in this thesis allows considering both inter-queue (inter- class) dependency as well as route inter-link end-to-end delay dependency in determining the delay operating points. To the best of our knowledge, no existing solution exploits both queue and route dependencies to find the class delay operating points, the class assignment, and the route configuration in multi-class DiffServ networks. A similar QoS-based class mapping that dynamically changes the end-to-end DiffServ class assignment to maintain end-to-end guarantees was proposed in [72]. Another DiffServ approach was presented in [73] to maintain the end-to-end QoS by allowing the promotion and demotion of DiffServ markings in links across the route. This approach limits the marking of DiffServ to 2 instead of 3. Both [72] and [73] consider class assignment independent of routing. Both approaches are based on the cost as seen by a given flow and do not consider the effect on other flows in the network.

3.3. Mathematical Formulation

Consider a network supporting a set $v \in V$ of QoS classes and described by the directed graph $G = (N, E)$ with a set $n \in N$ of nodes connected to each other using a set E of directional links; each link $e \in E$ is represented by a source-destination pair (s_e, t_e), where $s_e \in N$ and $t_e \in N$ are the source and destination nodes, respectively. Each link has a total capacity C_e, and link QoS class $v \in V$ carries an amount of traffic of $\lambda_{e,v}$. Traffic using class v on link e experiences an average delay $d_{e,v}$. A multi-class scheduling approach such as WFQ is used to serve traffic heading to link e. The average delay feasible region for a given link can be represented as a linear combination of the link delays weighted by the class loads, i.e,

$$\lambda_e d_e = \sum_v \lambda_{e,v} d_{e,v} \tag{3.1}$$

where λ_e is the total link traffic and d_e is the link average delay if this total traffic was served by a *single* queue.

The network supports a set I of end-to-end *trunks*; each trunk $i \in I$, denoted by the source-destination pair $(\hat{s}_i, \hat{t}_i)$, wishes to transfer a total amount of traffic γ_i from the source node[2] $\hat{s}_i$ to the destination node $\hat{t}_i$ using the set R_i of all possible *routes* with a required trunk delay limit of $\hat{D}_i$. The trunk i can select a subset of routes, $P_i \subset R_i$ where each route $p \in P_i$ is used to transport a partial amount of traffic, $\gamma_{i,p} > 0$ and is defined by a set $E_{i,p}$. Each element in $E_{i,p}$ represents a link-class pair $\{e, v\}$ that represents a QoS class v in link e. The formulation in this way is more general than traditional DiffServ QoS architecture since it enables a flow to attain different QoS class assignments on different links along the route. The end-to-end delay $\hat{d}_{i,p}$ of each route p is $\hat{d}_{i,p} = \sum_{\{e,v\} \in E_{i,p}} d_{e,v}$ and must be kept below a given delay requirement $\hat{D}_i$. It is desired to keep $\hat{d}_{i,p}$ as smaller than $\hat{D}_i$ as possible. As an indicator for each route p of trunk (or user) i, a *delay-margin penalty* function $U_{i,p}$ is defined as

$$U_{i,p} = \left[1 - \hat{d}_{i,p} / \hat{D}_i\right]^{-1} \tag{3.2}$$

The penalty function is used to construct an objective function for the overall network in the form of a weighted sum of the delay-margin penalty functions of all trunks,

$$U = \sum_{i \in I} \sum_{p \in P_i} \frac{\gamma_{i,p}}{\gamma_i} U_{i,p} \tag{3.3}$$

---

[2] The sign ^ is used to indicate an end-to-end variable

where the weight $\gamma_{i,p} / \gamma_i$ denotes the ratio of the partial amount of trunk i traffic, $\gamma_{i,p} > 0$ carried over the route p to the total amount of traffic, γ_i, of trunk i. Our objective is to find the optimum operating delay points for each link-QoS class, the sets of routes, $E_{i,p}$ and the corresponding allocated traffic amounts, $\gamma_{i.p}$, for all trunks in order to achieve the lowest possible weighted sum of the delay margin penalty functions, $U_{i,p}$. In other words, the *Multi-Class Delay-Margin Penalty Minimization* (MC-DMPM) problem is formulated as follows.

Find $E_{i,p}$, $\gamma_{i.p}$ and $d_{e,v}$ $\forall e \in E$, $\forall v \in V$, $\forall i \in I$ and $\forall p \in P_i$ to:

$$\min\left(\sum_{i \in I} \sum_{p \in P_i} \frac{\gamma_{i,p}}{\gamma_i} U_{i,p}\right) \tag{3.4}$$

while satisfying the following constraints:

$$\sum_{i \in I} \lambda_{e,v} < C_e \quad \forall e \in E, \forall v \in V \tag{3.5}$$

$$\gamma_i = \sum_{p \in P_i} \gamma_{i,p} \quad \forall i \in I \tag{3.6}$$

$$\sum_{t_e = t} \lambda_{e,i,v} - \sum_{s_e = t} \lambda_{e,i,v} = \begin{cases} -\gamma_i \text{if } t = \hat{s}_i \\ \gamma_i \ \text{ if } t = \hat{t}_i \\ 0 \ \text{ otherwise} \end{cases} \quad \text{where } \lambda_{e,i,v} = \sum_{(p \mid \{e,v\} \in E_{i,p})} \gamma_{i,p} \tag{3.7}$$

$$\hat{d}_{i,p} \leq \hat{D}_i, \forall i \in I, \forall p \in P_i, \quad \text{where } d_{i,p,v} = \sum_{e \in E_{m,p,v}} d_{e,v} \tag{3.8}$$

$$\lambda_e d_e = \sum_i \lambda_{e,v} d_{e,v} \quad \forall e \in E, \forall v \in V \tag{3.9}$$

$$d_{e,v}(w = 1) < d_{e,v} \quad \forall e \in E, \forall v \in V \tag{3.10}$$

where w is the weight of the weighted fair queue.

The formulation of the path delay margin penalty function in the form of (3.2) has many advantages. The penalty function goes to infinity as the path delay approaches the delay requirement specified by the end-to-end delay constraint (3.8). Using admission at network edges to prevent accepting calls as the penalty approaches ∞ would ensure that the function is only valid in the range form $\hat{d}_{i,p} = 0$ to $\hat{d}_{i,p} = \hat{D}_i$ and hence will always remain in the delay feasible region.

When the traffic demand for each service is defined, the delay margin penalty function will maximize the number of accepted users compared with network utility function maximization. Maximizing the utility function constrained by the QoS requirement will achieve a higher network utility in the presence of unlimited traffic demand. The network utility maximization is achieved by favoring flows with less demanding QoS requirements. However, for a defined traffic demand, the number of flows with less demanding QoS requirements is limited. Favoring them may lead to decreasing the network load because it may lead to rejecting flows with demanding QoS requirements. Although the delay margin cost function focuses on equally enhancing the end-to-end QoS compared with its requirement, it implicitly considers the network utility. Because delay is proportional to network load, maximizing the delay margin is equivalent to decreasing the load between end-to-end nodes. The delay margin penalty function presents the advantage of equally considering the delay requirements of all users in the network load minimization. For any end-to-end delay requirement $\hat{D}_i < \infty$, the capacity constraint (3.5) is also implied in the penalty function formulation, since any link delay $d_{e,i}$ would go to infinity as the λ_e approaches C_e and, hence, will cause $\hat{d}_{i,p}$ to approach $\hat{D}_i$. Using this approach to minimize the network load enables the accommodation of more QoS users and hence increases the network utility in the presence of defined traffic demand and end-to-end QoS minimization.

Constraints (3.6), and (3.7) are required in any flow optimization problem as they guarantee that all the traffic sent by a given trunk is received at the trunk destination and that all traffic input to a node is equal to the traffic output except at the source and destination. The multiplier $\gamma_{i,p} / \gamma$ provides continuity in the objective function(3.4). Constraints (3.9) and (3.10) were added to the problem formulation to represent the general work-conserving scheduler. The equality constraint in (3.9) represents the inter-relation between the average delays of different classes, which is introduced by the work conserving scheduler. It ensures that the class delays are chosen from the linearly related feasible set characterized by (3.9). Constraint (3.10) ensures that any chosen delay is larger than the delay received when this class has priority over all other classes. Without constraints (3.9) and (3.10), the problem can be treated as an unconstrained nonlinear optimization. When (3.9) and (3.10) are added, the problem is transformed into a constrained nonlinear optimization problem. Investigation of the function convexity in the presence of the delay operating point adjustment capability shows that the function represents a convex hull that is interrupted by concave areas. These concave areas result when a trunk splits its load between two classes at the same link. In this case, the minimum of U falls on the edges of this concave area, which corresponds to the case when all the trunk traffic is assigned to only one class.

Convexity of delay-margin penalty function: For (3.2) to be convex over a set of delay feasible load points $\gamma_{i,p}$, two conditions have to be satisfied, i.e., the domain of feasible delay loads $\gamma_{i,p}$ has to be convex and $\nabla^2 U$ has to be positive for any value in the feasible domain.

Convexity of feasible delay domain: For the domain of feasible delay loads $\gamma_{i,p}$ to be convex, the load points on the line $\gamma = \alpha\gamma^{(1)} + (1-\alpha)\gamma^{(2)}$ (where $0 \le \alpha \le 1$) connecting any two delay feasible load points $\gamma^{(1)}$ and $\gamma^{(2)}$ have to lie within the feasible set. Points on that line can be interpreted as a load configuration (i.e., a combination of both feasible load points reduced by factors α and $1-\alpha$). Since delay is an increasing function of load, then each component is still guaranteed to be within the feasible region, and hence the domain of feasible delay $\gamma_{i,p}$ is convex. From (3.11) each component in $\nabla^2 U$ can be defined as:

$$
\begin{aligned}
\frac{\partial^2 U_{i,p}}{\partial\gamma_{n,q}\partial\gamma_{w,z}} = & \frac{2\gamma_{i,p}}{\gamma\hat{D}_{i,p}(1-d_{i,p}/\hat{D}_{i,p})^3}\frac{\partial\hat{d}_{i,p}}{\partial\gamma_{n,q}}\frac{\partial\hat{d}_{i,p}}{\partial\gamma_{w,z}} + \\
& \frac{1}{\gamma\hat{D}_{i,p}(1-d_{i,p}/\hat{D}_{i,p})^2}\left(\frac{\partial d_{i,p}}{\partial\gamma_{w,z}}\frac{\partial\gamma_{i,p}}{\partial\gamma_{n,q}} + \frac{\partial\hat{d}_{i,p}}{\partial\gamma_{w,z}}\frac{\partial\gamma_{i,p}}{\partial\gamma_{w,z}}\right) + \\
& \frac{1}{\gamma\hat{D}_{i,p}(1-d_{i,p}/\hat{D}_{i,p})}\left(\frac{\gamma_{i,p}}{\hat{D}_{i,p}}\frac{\partial^2\hat{d}_{i,p}}{\partial\gamma_{n,q}\partial\gamma_{w,z}} + \frac{\partial^2\gamma_{i,p}}{\partial\gamma_{n,q}\partial\gamma_{w,z}}\right)
\end{aligned}
\qquad (3.11)
$$

For a *single-class* network $I = \{1\}$, delay is a convex monotonically increasing function of γ [92]. Hence, $\partial \hat{d}_{i,p} / \partial \gamma_{w,z}$ $\partial \hat{d}_{i,p} / \partial \gamma_{n,q}$, $\partial^2 \hat{d}_{i,p} / (\partial \gamma_{n,q} \partial \gamma_{w,z})$ are all positive. On the other hand, $\left(1 - \hat{d}_{i,p} / \hat{D}_{i,p}\right)$ is positive for all delay feasible load points, hence (3.11) is always positive within the feasible region. This means $\nabla^2 U$ is always positive, i.e., the penalty function is convex with the domain of feasible delay. On the other hand,

$$\frac{\partial U}{\partial \alpha_{i,p,e}} = \frac{\gamma_{i,p}}{\gamma \hat{D}_i\left(1-d_{i,p1}/\hat{D}_i\right)} + \left(\frac{\alpha_{i,p,c}\gamma_{i,p}}{\left(\gamma \hat{D}_i\left(1-d_{i,p1}/\hat{D}_i\right)\right)^2} + \sum_{\substack{i,p|\{e,1\}\in Ei,p \\ \{i,p\}\neq\{v,q\}}} \frac{\gamma_{v,q}}{\left(\gamma \hat{D}_n\left(1-d_{n,q}/\hat{D}_n\right)\right)^2} \right) \frac{\partial d^*_{e,1}}{\partial \alpha_{i,p,e}}$$

$$- \frac{\gamma_{i,p}}{\gamma \hat{D}_i\left(1-d_{i,p2}/\hat{D}_i\right)} + \left(\frac{(1-\alpha_{i,p,e})\gamma_{i,p}}{\left(\gamma \hat{D}_i\left(1-d_{i,p2}/\hat{D}_i\right)\right)^2} + \sum_{\substack{n,q|\{e,1\}\in En,q \\ \{i,p\}\neq\{n,q\}}} \frac{\gamma_{n,q}}{\left(\gamma \hat{D}_v\left(1-d_{n,q}/\hat{D}_n\right)\right)^2} \right) \frac{\partial d^*_{e,2}}{\partial d^*_{e,1}} \frac{\partial d^*_{e,1}}{\partial \alpha_{i,p,e}}$$

Where $$\frac{\partial d^*_{e,2}}{\partial d^*_{e,1}} = -\frac{\lambda_{e.1}}{\lambda_{e.2}} = -\frac{\alpha_{i,p,e}\gamma_{i,p} + \lambda_{e,1}^{-\{i,p1\}}}{(1-\alpha_{i,p,e})\gamma_{i,p} + \lambda_{e,2}^{-\{i,p2\}}},$$

and $$\lambda_{e,1}^{-\{i,p1\}} = \sum_{\substack{n,q|\{e,1\}\in En,q \\ \{i,p1\}\neq\{n,q\}}} \gamma_{v,q},\quad \lambda_{e,2}^{-\{i,p2\}} = \sum_{\substack{n,q|\{e,2\}\in En,q \\ \{i,p2\}\neq\{n,q\}}} \gamma_{n,q}$$

then $\dfrac{\partial^2 U}{\partial \alpha_{i,p,e}^2} = H1 + H2$ where

$$H1 = \frac{\gamma_{i,p}}{\left(\gamma \hat{D}_i\left(1-d_{i,p1}/\hat{D}_i\right)\right)^2} + \left(\frac{\alpha_{i,p,c}\gamma_{i,p}}{\left(2\gamma \hat{D}_i\left(1-d_{i,p1}/\hat{D}_i\right)\right)^3} + \sum_{\substack{i,p|\{e,1\}\in Ei,p \\ \{i,p\}\neq\{n,q\}}} \frac{\gamma_{n,q}}{\left(2\gamma \hat{D}_i\left(1-d_{n,q}/\hat{D}_i\right)\right)^3} \right) \frac{\partial^2 d_{e,1}}{\partial \alpha_{i,p,e}^2}$$

$$
\begin{aligned}
H2= & -\frac{\gamma_{i,p}}{\gamma \hat{D}_i\left(1-d_{i,p2} / \hat{D}_i\right)^2} \\
& +\left(\frac{(1-\alpha_{i,p,e})\gamma_{i,p}}{2\gamma \hat{D}_i\left(1-d_{i,p2} / \hat{D}_i\right)^3}+\sum_{\substack{n,q \mid \{e,1\} \in En,q \\ \{i,p\} \neq \{n,q\}}} \frac{\gamma_{n,q}}{2\gamma \hat{D}_n\left(1-d_{n,q} / \hat{D}_n\right)^3}\right) \frac{\partial d_{e,2}^*}{\partial d_{e,1}^*} \frac{\partial d_{e,1}^*}{\partial \alpha_{i,p,e}} \\
& +\left(\frac{(1-\alpha_{i,p,e})\gamma_{i,p}}{\gamma \hat{D}_i\left(1-d_{i,p2} / \hat{D}_i\right)^2}+\sum_{\substack{n,q \mid \{e,1\} \in En,q \\ \{i,p\} \neq \{n,q\}}} \frac{\gamma_{n,q}}{\gamma \hat{D}_n\left(1-d_{n,q} / \hat{D}_n\right)^2}\right)\left(\frac{\partial d_{e,2}^*}{\partial d_{e,1} \partial \alpha_{i,p,e}} \frac{\partial d_{e,1}^*}{\partial \alpha_{i,p,e}}+\frac{\partial d_{e,2}^*}{\partial d_{e,1}^*} \frac{\partial^2 d_{e,1}^*}{\partial \alpha_{i,p,e}^2}\right)
\end{aligned}
$$

$H1$ is always positive, while $H2$ is negative since $\partial d_{e,2}^* / \partial d_{e,1}^*$ and $\partial d_{e,2}^* / (\partial d_{e,1} \partial \alpha_{i,p,e})$ are negative. When $|H2| > |H1|$, $\partial^2 U / \partial \alpha_{i,p,e}^2 < 0$ and hence the function relating U to moving a trunk traffic from one class to another is *not* convex in this region. In other words, in this case, for a given route loading of a trunk, the minimum of U is minimum when traffic of all the routes is assigned to only one class, i.e., the minimum value will fall on either side of the function.

3.4. Centralized TE Schemes

The centralized approach in this thesis decomposes the optimization problem in (3.4) into three sub-problems. The first one (to be discussed in 3.4.1) focuses on *route configuration.* It aims to find the best set of *routes* for each *trunk* and the corresponding traffic loading for each *route.* Based on this resulting route configuration, the second sub-problem (to be discussed in Section 3.4.2) deals with the *link class delay assignment.* It searches for the optimal set of *delay operating points* for all the *links.* The third sub-problem (to be discussed in Section 3.4.3) aims to overcome the concave areas that may arise as discussed in Section 3.3. In *single-class* networks, centralized offline route configuration is the only required algorithm.

3.4.1. Centralized offline route configuration:

The centralized offline route configuration is a gradient-based multi-class nonlinear optimization that operates on the convex region of the objective function. The algorithm, summarized in Figure 3-2, iteratively moves the trunk traffic from routes/classes with high objective function increase rates to routes/classes with lower increase rates, and hence gradually descends to the optimal routing configuration. The iterations are repeated until the percentage decrease in the delay margin penalty function $\Delta = 1 - U^{(k+1)} / U^{(k-i)}$ falls below a given threshold η. The value of η represents the smallest delay margin enhancement that is required in the system. This value will depend on the delay requirements in the network and the desired algorithm speed. A small value of η (e.g 10^{-5}) will ensure a network configuration that is close to the optimal value but will require a big number of iterations and hence will slow down the algorithm. On the other hand a large value of η (e.g 10^{-2}) will speed the algorithm but may not decrease the delay margin minimization and hence may decrease the number of users accepted in the network.

When there is change in traffic of *trunk* n assigned to *route* $E_{n,q}$, the delay-margin penalty of traffic $\gamma_{m,p}$ (of trunk m on *route* $E_{i,p}$) will change, if both $E_{i,p}$ and $E_{n,q}$ share at least one link-class pair $\{e,v\}$. The impact of this change can be represented by the overall rate of change in the objective function,

$$L_{n,q} = \frac{\partial U}{\partial \gamma_{n,q}} = \sum_i \left(\frac{1}{\gamma_i} \sum_p \left(U_{i,p} \frac{\partial \gamma_{i,p}}{\partial \gamma_{n,q}} + \gamma_{i,p} \frac{\partial U_{i,p}}{\partial \gamma_{n,q}} \right) \right),$$

$$\text{where} \quad \frac{\partial U_{i,p}}{\partial \gamma_{n,q}} = \frac{U_{i,p}^2}{\hat{D}_i} \frac{\partial \hat{d}_{i,p}}{\partial \gamma_{n,q}} \quad \text{and} \quad \frac{\partial \hat{d}_{i,p}}{\partial \gamma_{v,q}} = \sum_{e \in (E_{i,p} \cap E_{n,q})} \frac{\partial d_{e,v}}{\partial \gamma_{n,q}}$$

where $\partial \gamma_{i,p} / \partial \gamma_{n,q}$ is 1 if $E_{i,p} = E_{n,q}$ and zero otherwise. Similarly $\partial d_{e,v} / \partial \gamma_{n,q}$ is non-zero only when $e \in E_{n,q}$. Re-arranging terms and replacing $\partial \gamma_{n,q}$ by $\partial \lambda_{e,v}$ (since at any node $\lambda_{e,v} = \sum_n \sum_q \ (\gamma_{n,q} \mid \{e,v\} \in E_{n,q})$), $L_{n,q}$ can be rewritten as

(a) Route Configuration:

while $(\Delta < \eta)$
for $(\forall i \in I)$
for $(\forall p \in R_i)$ $L_{i,p,v} = \partial U \;/ \partial \gamma_{i,p,v}$ *end*
for $E^*_{i,p} = (E_{i,p} \mid L_{i,p} = \min_{p \in R_i}(L_{i,p}))$
$P_i = P_i \cap p$
$\alpha^* = \arg\min_{\alpha}(\sum_i \sum_p U_{i,p}(\alpha))$

$$\textit{where } \gamma^{(k+1)}_{i,p} = \begin{cases} \gamma_{i,p} + \alpha(\gamma^{(k)}_i - \gamma^{(k)}_{i,p}) & \text{if } E_{i,p} = E^*_{i,p} \\ (1-\alpha)\gamma^{(k)}_{i,p} & \text{if } E_{i,p} \neq E^*_{i,p} \end{cases}$$

$U^{(k+1)} = \sum_i \sum_p U_{i,p}(\alpha^*))$
end for
Link Class Delay Assignment
Route Traffic Promotion/Demotion
$\Delta = 1 - U^{(k+1)} / U^{(k-i)}$
end while

Figure 3-2: Route Configuration

$$L_{n,q} = \frac{1}{\gamma_i}\left[U_{n,q} + \sum_{\{e,j\} \in (E_{n,q})} l_{e,j} \right] \tag{3.12}$$

$$\text{where } l_{e,j} = \sum_{V} \frac{\partial d_{e,v}}{\partial \lambda_{e,j}} \psi_{e,v}, \quad \text{and} \qquad \psi_{e,v} = \sum_{m} \sum_{(p \mid \{e,v\} \in (E_{i,p}))} \gamma_{i,p} \frac{U^2_{i,p}}{\hat{D}_i} \tag{3.13}$$

$l_{e,j}$ in the second term in Equ. (3.12) is the change in the network objective function with respect to the change of class j traffic on link e. It can be used to represent the length of a given class on a given link and hence provides a partial decomposition of the problem on a per-class per-link basis. In calculating the route length using Equ. (3.12), each class can be considered as a different link with its own length in the single-class case. The form of $l_{e,j}$ has its advantages in real-life implementation as indicated in [50] for the single-class case. The main advantage in the multi-class case is that each link can characterize the load dependencies between its classes through the calculation of its own $\partial d_{e,j} / \partial \lambda_{e,v}$. These values can be calculated by each network class based on the monitoring and characterization of its own traffic.

The complexity of this algorithm is similar to that of a gradient-based algorithm, e.g., flow deviation. Flow deviation converges in $O(\phi^2 e^5 \varepsilon^{-3} i)$ of route calculations [88], where ϕ is the width, ε is the performance and we abuse the notation and use e to represent the number of links and, i to represent the number of trunks. Because at each link, the number of choices equals the number of classes, i, this algorithm is expected to converge in $O(\phi^2 \left[ev\right]^5 \varepsilon^{-3} i)$. The main complexity of this algorithm comes from route calculation. Although a minimum value of each term in $L_{n,q}$ can be calculated using a shortest path approach, their sum requires the use of min-cost routing that is constrained by the value of $\hat{D}_{n,q}$. In the centralized approach in this paper we assume the use of pre-computed routes [94] to speed up this evaluation. We also present an approximation in Section 3.5

3.4.2. Link class delay assignment

Based on the allocation of routes, route loads, and route class assignment provided by the *route configuration* algorithm, this algorithm searches for an optimal set of *delay operating points* for all the *links*. The difficulty of the problem arises from the fact that constraints (3.9) and (3.10) are not implied by the objective function and hence the problem has to be solved as a constrained non-linear optimization problem. Figure 3-3 summarizes the proposed solution that uses a linear approximation of the function at each iteration to find a solution to the resulting constrained linear optimization problem. Line search is then used to find a lower-cost point on the nonlinear function using the linear approximation. The linear approximation is based on the function gradient represented by

$$R_{e,j} = \frac{\partial U}{\partial d_{e,j}} = \sum_i \sum_p \frac{\gamma_{i,p}}{\gamma_i} \frac{\partial U_{i,p}}{\partial d_{e,j}} = \sum_i \psi_{e,v} \frac{\partial d_{e,v}}{\partial d_{e,v}} \qquad (3.14)$$

where $\psi_{e,v}$ is defined by (3.13). To calculate $R_{e,v}$ at the link level, each link needs to keep track of the value of $\psi_{e,v}$ for each class, and each class needs to keep track of $\partial d_{e,v} / \partial d_{e,j}$ with respect to all other classes j on the same link. The value $\partial d_{e,v} / \partial d_{e,j}$ depends on the multi-class queue scheduling approached used in the network. This value can be calculated by the link traffic engineering module based on its own characterization of its input traffic and its current queue configuration. These values can be periodically conveyed to the traffic-engineering server to be used in the optimization, or they can be used locally in between optimization to provide a local approximation to the centralized link class assignment. In the simple two-queue case, $\partial d_{e,v} / \partial d_{e,j}$ can be obtained from (3.9). The optimal values of link class $d^*_{e,v}$ delays can be obtained by iteratively using line search techniques to solve the nonlinear optimization for all combination of e and v and constrained by (3.9) and (3.10). The calculation of $R_{e,j}$ requires a maximum i trunk updates and *(v-1)* calculation of $\partial d_{e,v} / \partial d_{e,j}$ where v is the number of classes. The calculation of $\tilde{\mathbf{d}}$ is a linear optimization problem that has delay constraint equalities up to $i\bar{p}$, where i is the number of trunks and $\bar{p}$ is the maximum number of routes per trunk. Because of the work conserving properties of the link schedulers, the system has e equalities, where e is the number of links. Each equality describes the relation between link class delays. For a simplex method, the convergence time for the calculation of $\tilde{\mathbf{d}}$ is an exponential function of ($i\bar{p} + e$).

(b) Link Class Delay Assignment:

while $\eta > \varepsilon$

$$\tilde{U} = U^{(k)} + \sum\sum (\tilde{d}_{e,j} - d^{(k)}_{e,j}) R_{e,j}$$

$$\tilde{\mathbf{d}} = \arg\min_{\mathbf{d}^{(k)}} \left(\sum_e \sum_j \tilde{d}_{e,j} R_{e,j} \Big|_{\mathbf{d}^{(k)}} \right)$$

$$\mathbf{d}^{(k+1)} = \alpha\tilde{\mathbf{d}} + (1-\alpha)\mathbf{d}^{(k)}$$

$$\alpha = \arg\min_{\alpha} \left[U(\alpha\tilde{\mathbf{d}} + (1-\alpha)\mathbf{d}^{(k)}) \right]$$

Calculate $U^{(k+1)}$ *using (3.2) and*

$$\eta = \left| (U^{(k)} - U^{(k+1)}) / U^{(k)} \right|$$

end while

Figure 3-3: Link Class Delay Assignment

3.4.3. Route traffic promotion/demotion:

This algorithm aims to move the optimization process from one convex area to the other by examining the routes (of different trunks) one by one as shown in Figure 3-4. For each link on each route of the same trunk, it checks if further reduction in the objective function can be achieved by moving the route traffic on this link to a higher or lower delay class. If a better class is found, the trunk route is promoted or demoted and the objective function is updated accordingly.

The traffic promotion/demotion step in the algorithm has two features: (i) the move of traffic between classes is done per-flow, which provides an opportunity for distributed end-to-end implementation; (ii) the move between classes is done over the same established path, which provides a means for collecting information about the cost of other classes along the route links using protocols such as RSVP-TE [25].

(c) Route Traffic Promotion/Demotion:

for $(\forall e \in E)$ & $(\forall v \in V)$
  for $(\forall \{i,p\} \mid \{e,v\} \in E_{i,p})$
    for $(\forall j \in V \mid v \neq j)$
    $\tilde{E}_{i,p} = E_{i,p} \cup \{e,j\} / \{e.i\}$
    $\tilde{\lambda}_{e,v} = \tilde{\lambda}_{e,v} - \gamma_{i,p}$
    $\tilde{\lambda}_{e,j} = \lambda_{e,j} + \gamma_{ip}$
     if $U(\{\tilde{\lambda}_{e,j}, \tilde{\lambda}_{e,i}\}) < U(\{\lambda_{e,j}, \lambda_{e,v}\})$
      $E_{m,p} = \tilde{E}_{i,p}$
     end if
    end for
  end for
end for

Figure 3-4: Route Traffic Promotion/Demotion

3.5. Online approximations

3.5.1. Distributed online routing approximation:

In this thesis we investigate two levels of online routing approximations. The first approach is a flow-by-flow approach. For each arriving flow, the first approximation makes use of $L_{i,p}$ to find $\tilde{E}^*_{i,p}$ the optimal path for an arriving trunk i flow at its time of arrival. By doing that the flow is assigned to the route that would cause the least increase in the delay margin penalty function. In networks where flows have sojourn time, older flows will leave the network and newer flows will be assigned to the routes that would decrease the delay margin penalty function. This will cause an effect that is similar to traffic shifting that is done in the centralized approach. The first term in (3.12) prevents the distributed implementation of this routing approach. The first term in (3.12) is an end to end term that represents the end-to-end delay constrains. This complicates the routes search approach and requires the use of constrained routing approaches. For the rest of the thesis we refer to this approach as DRPM-Exact Route length calculations (D-ER).

To enable a more distributed implementation of the D-ER approach that can make use of vector routing tables, we use an approximation for the calculation of $L_{i,p}$. The approximation is based on the observation that the smallest value of $L_{i,p}$ occurs either at the path that minimizes the first term in (3.12) by minimizing the value of $\hat{d}_{i,p}$ or by minimizing the second term in (3.12). The value of $\hat{d}_{i,p}$ can be minimized using a shortest path routing that uses the delay of each class delay $d_{e,v}$ as the link length. The second term can in (3.12) can be minimized using a shortest path approach that uses $l_{e,j}$ as the link length. Each of these two shortest paths can be implemented using vector routing tables. For both paths, the Dijstra [30] algorithm can_be used for the calculations. This means that each node will maintain two routing tables one based on $d_{e,v}$ as the link length and the other based on $l_{e,j}$ as the link length. The distributed approach in this thesis assumes the use of route selection messages such the path message in RSVP-TE [25]. Two of these messages are sent for each arriving flow. The first one is routed searches for the route with the smallest sum of $d_{e,v}$, and has to record the value of $l_{e,j}$ along way, while the second searches for the route with the smallest sum of $l_{e,v}$ and has to record $d_{e,v}$ along the way.

At any intermediate node if several paths with the same smallest delay exist, the one with the smallest sum of $l_{e,v}$ is chooses and similarly, if several shortest paths based on $l_{e,v}$ are found, then the one with the smallest delay is chosen. For the rest of the thesis we refer to this approach as DRPM-Approximate (D-AR). The destination node receives both messages for the two chosen shortest paths and uses them to calculate the exact value of $L_{i,p}$ for each path. The destination then chooses the path with the smaller value of $L_{i,p}$. A path establishment message such as RSVP-TE [25] Resv message is sent on the chosen path reverse direction to establish the connection. This distributed implementation requires each node to keep and update 2 vector routing tables per class, one recording the shortest path to any node in the network in terms of delay and the other recording the shortest path in terms of $l_{e,v}$. These vector routing tables can be built and maintained using OSPF. It also requires the presence of path-established protocols such as RSVP-TE or CR-LDP. This reduces the complexity of the routing approach to two times that of shortest path routing multiplied by the number of classes.

3.5.2. Online class delay assignment:

The online delay class assignment aims to minimize the cost function by optimizing the delays at each link such as to set the gradient $R_{e,j}$ as close to zero as possible for all classes j on each link. Because of the work conserving properties of the link schedulers, only the first term $\psi_{e,j}$ $\partial d_{e,v} / \partial d_{e,j}$ in the $R_{e,j}$ will be positive and all other terms will be negative. Increasing the delay of a given class increases the objective function of flows going through this class and decreases the objective function of flows going through other classes. A negative value of $R_{e,j}$ means that increasing the delay of class j will decrease the overall network cost. The algorithm is triggered by periodic updates to $\psi_{e,v}$. It works by calculating the value of $R_{e,j}$ after each update, and the delay of the class with the smallest negative $R_{e,j}$ class is increased by small value α.

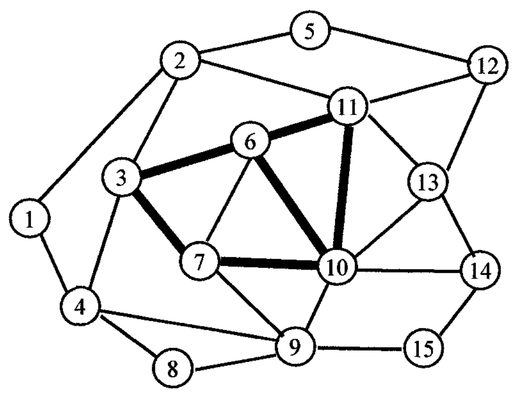
3.6. Single Class Delay Margin based TE

3.6.1. Simulation scenario

To evaluate the performance of the proposed approach, we will start first by evaluating the single class version of the algorithm versus other existing single class approaches. The objective is to confirm that the use of delay satisfaction routing can enable the network to accommodate more users even in a single class network. In order to examine the impacts of more realistic bursty traffic with varying loads, each call is modeled as a Poisson traffic source with an average rate γ_c. The trunk total traffic at any point in time is equal to the sum of the traffic generated by calls admitted to this trunk. In this chapter we assume that this sum is Poisson distributed. Based on the Poisson traffic characteristics and the independence assumption [83], the average delay for priority queues fed with Poisson packet arrivals [92] is used to numerically evaluate the average delay at each class in a given link, and hence evaluate the end-to-end delay for different trunks.
In the single class evaluation we use the network topology Net1 that is shown in Figure 3-5. Net1 uses the same example network topology used in [93]. The dark links have a capacity of 4800 units and the other links have 1200 units. This section uses the same call traffic characteristic and source destination pairs for trunk requirements as in [93], but each source destination pair in [93] is used to create two different trunks with different delay requirements. This is intended to simulate QoS networks where trunks are defined by their QoS requirements as well as their source-destination pair. It is also used to see if the proposed approaches will be able to map trunks with different delay requirements in the appropriate end-to-end priority class. In this example, the value of γ_c is uniformly distributed between 1 and 4. The traffic and QoS requirements of Trunk/Calls in Net1 are shown in Table 3-3. Net 1 is used in the evaluation of both the single class and multi-class performance evaluation.

Following the same approach as [93], we consider both static and dynamic cases based on the call arrival behaviour. In the *static* case, at each simulation step, a trunk is randomly chosen to generate a call. If the call is admitted to the trunk, the trunk total traffic is increased by γ_c, otherwise the amount of unaccepted traffic is increased by γ_c. Once admitted, the call stays until the end of the simulation. In other words, the traffic offered to the network will increase gradually with the arrival of calls. As the traffic increases, end-to-end delay will increase, causing some trunks to approach their QoS limits and hence preventing them from accepting more new calls. A trunk operating close to its delay requirement will also prevent other trunks sharing links with it from accepting calls, as this will cause its own delay violation. When the centralized approach is used, the optimization process is re-run whenever a new call is accepted. The performance of an approach in this simulation is measured by its capability to accept more calls in the network (or, equivalently, to decrease the amount of unaccepted traffic) with the increase in offered traffic.

The admission of a new call with an average traffic rate γ_c is decided as follows. The call is assigned to the route/class with the smallest value of $L_{i,p,v}$ (or its approximation). Admitting the



(a) Net1 (from [93])

Figure 3-5: Example Network Topologies

Table 3-3: Trunk Requirements for Net 1

trunk			Call Requiremen		trunk			Call Requiremen	
i	$\hat{s}_i$	$\hat{t}_i$	γ_i	D_i	i	$\hat{s}_i$	$\hat{t}_i$	γ_i	D_i
1	1	13	1-4	.004	5	1	13	1 -4	.008
2	5	9	1-4	.006	6	5	9	1-4	.012
3	2	4	1-4	.003	7	2	4	1-4	.006
4	5	15	1-3	.005	8	5	15	1 4	.010

call in the network is expected to increase the delay on this route links. Using a multi scale characterization of the arriving call traffic characteristic, this expected delay increase is calculated using the analytical technique in chapter 4. The call is admitted in the network if its delay requirement as well as the delay requirements of established calls can be guaranteed despite the estimated link delay increase.

3.6.2. Single class static

For the static case, we consider the fraction of rejected calls. The results for various routing algorithms are summarized in Figure 3-6. Consider a negligible traffic rejection ratio of much lower than 0.1%., the minimum-hop has the worst performance with about 1150 traffic units. It is interesting to observe that the performance of MIRA is degraded when delay constraints are

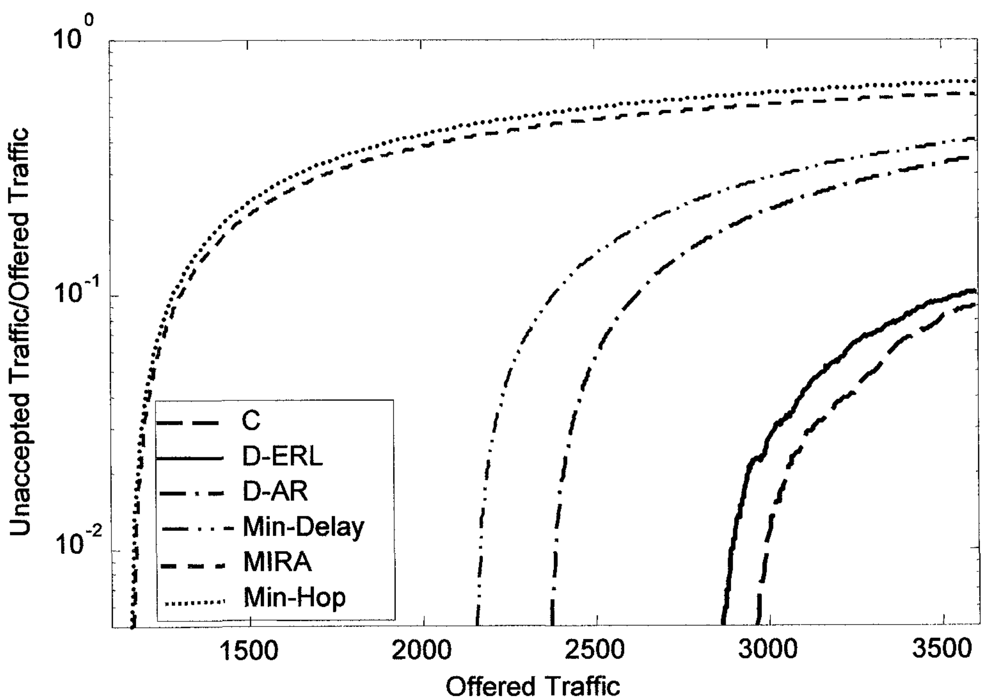


Figure 3-6: Fraction of rejected calls versus the number of calls

introduced, and gets very close to that of minimum-hop routing. As the min-delay routing algorithm aims to find the minimum-delay routes and delay is related to the link residual bandwidth, it outperforms the minimum-hop and MIRA with about 2150 accepted traffic units (or an increase of more than 85%) for a negligible traffic rejection ratio.

The DRPM approaches provide a superior performance with 2360, 2835, and 2930 accepted traffic units (or an increase of 105%, 146% and 154% as compared to the minimum-hop and MIRA) for Distributed DRPM-Approximate (D-AR), Distributed DRPM-Exact (D-ER) and Centralized DRPM (C), respectively. The online Distributed DRPM-Exact has a performance very close to that of the centralized DRPM with lower computational complexity. The small decrease in the amount of traffic is mainly due to the fact that the distributed version cannot change the route selection for connections that arrived earlier. Although the optimality at the route selection may remain at the call arrival time, it is affected by the presence of new s-t pairs in the network. The centralized computation has more global view of the network, and hence has the advantage of being able to divert established calls to new routes that may provide a better overall network performance.

3.6.3. Single class dynamic Case:

For the dynamic case, we compare the performance of the distributed DRPM, min-delay and min-hop routing algorithms in terms of accepted traffic versus time over a period of 100,000 time units, as shown in Figure 3-7. The results confirm the ability of both approximations to increase the amount of traffic accepted in the network. Unlike the static case, the approximate calculation of D-AR performs better in the dynamic case. This is mainly due to the dynamics of users. These dynamics allow the network to correct its previous choices of routes once calls using these routes leave the network. This gives a similar effect to the choice of α in the DRPM algorithm.

3.7. Delay Margin TE based on WFQ schedulers

3.7.1. Example scenario

In the multi-class networks, to make sure that results are not biased by the choice of the network in [93], we use another randomly generated topology, Net2 (shown in Figure 3-8). The traffic and QoS requirements of Trunk/Calls in Net1 are shown in Table 3-4.

The example transports 21 pairs of trunks. Each pair has the same source and destination but different delay requirements (e.g., 3ms, 6ms). Calls of equal traffic requirements are randomly generated using a uniform distribution over the trunks. The source destination pairs for trunks in Net2 are shown in Figure 3-8. In order to examine the impact of more realistic bursty traffic with varying loads, each call is modeled as a Poisson traffic source with at an average rate γ_c. The trunk total traffic at any point in time is the sum of the Poisson-distributed traffic generated by calls admitted to this trunk. Generated calls have exponentially distributed inter-arrival time and

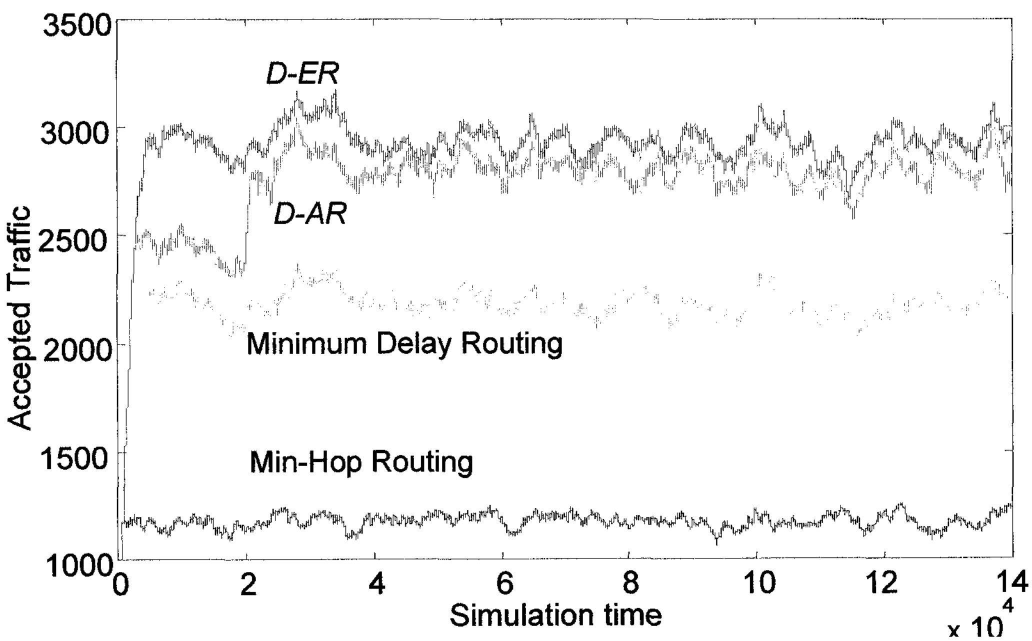


Figure 3-7: Single class accepted traffic versus time (dynamic)

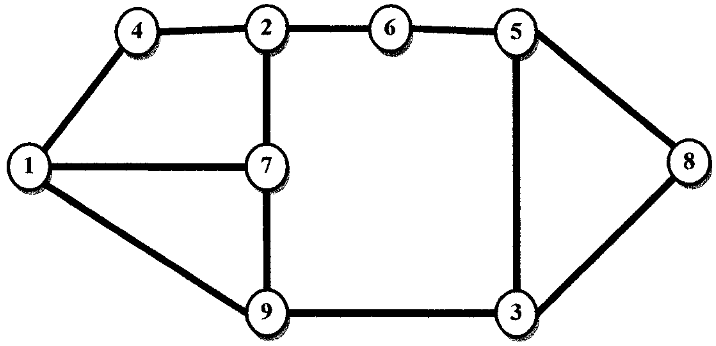


Figure 3-8: Example Network Topology (NET2)

Table 3-4: Trunk Requirements for Net 2

i	$\hat{s}_i$	$\hat{t}_i$	i	$\hat{s}_i$	$\hat{t}_i$	i	$\hat{s}_i$	$\hat{t}_i$	i	$\hat{s}_i$	$\hat{t}_i$	i	$\hat{s}_i$	$\hat{t}_i$	i	$\hat{s}_i$	$\hat{t}_i$	i	$\hat{s}_i$	$\hat{t}_i$
1	1	6	4	2	4	7	3	5	10	5	1	13	7	2	15	7	1	19	9	7
2	1	3	5	2	3	8	4	2	11	5	4	14	7	8	16	8	5	20	9	6
3	1	4	6	3	5	9	4	2	12	6	1	18	8	3	17	8	7	21	9	5

call duration. The average call duration is 250 and 300 times the mean inter-arrival for Net1 and Net2, respectively. In the simulation we use a two queue WFQs for scheduling traffic at the links. We consider the independence assumption [83] to allow easy aggregation of the traffic. The scenario uses the same connection admission and call generation methods as that used in Section 3.6.1. Because of the absence of an accurate performance analysis for WFQ, we use an empirical approximation of the delay function where

$$d_1 = \begin{cases} (C - w_2\lambda - \lambda_1)^{-1} & \text{if}\,(d_1 < (C-\lambda)^{-1}) \\ (\lambda / \lambda_1)(C-\lambda)^{-1} - (\lambda_2 / \lambda_1)(C - (1-w_2)\lambda - \lambda_2)^{-1} & \text{if}\,(d_1 > (C-\lambda)^{-1}) \end{cases} \tag{3.15}$$

To examine the algorithms we will gradually introduce them to evaluate the effect of each of them. We will start with a network that uses min-hop or min-delay algorithms. As a starting point we assume that trunks are assigned to end-to-end QoS classes corresponding to their delay. Link weights are set to 0.6 and 0.4, hence creating a higher and lower delay classes. We will then introduce each algorithm separately and then combine them to show the effect of them working together.

3.7.2. WFQ Static case

Figure 3-9 and Figure 3-10 show the results for min-hop and min-delay routing. They show that when Class Assignment (CA) is introduced, it increases the amount of accepted traffic for both min-hop and min-delay routing. This is mainly due to the added choices along the route and the ability to distribute resources along the route. This is done by promoting the QoS of more LSPs at more congested links and demoting them at highly loaded links. Demotion may also be used to compensate for high QoS that is gained by sharing certain classes with connection that have a stringent QoS requirement. When delay operating point assignment (DA) is introduced, it presents a major increase in the amount of accepted traffic. This choice represents an adaptation to the end-to-end trunk QoS requirements and network topology. As expected, when both are combined a better performance is achieved.

Figure 3-11 shows the results when the delay margin route configuration is introduced. It

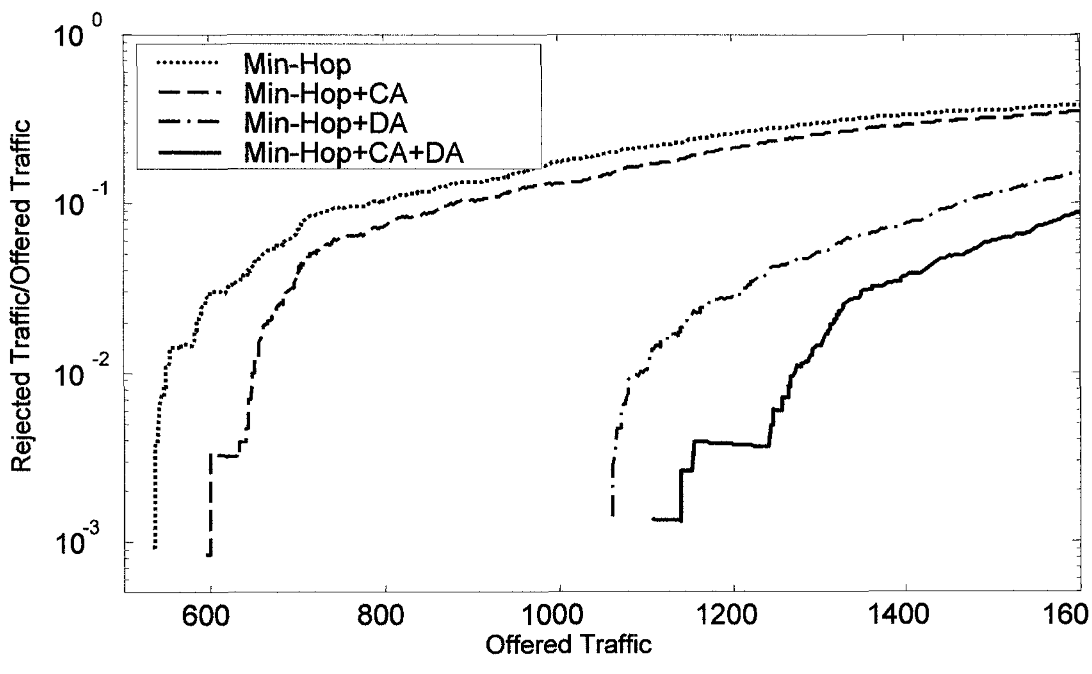


Figure 3-9: Class and delay assignment applied to Min-hop routing

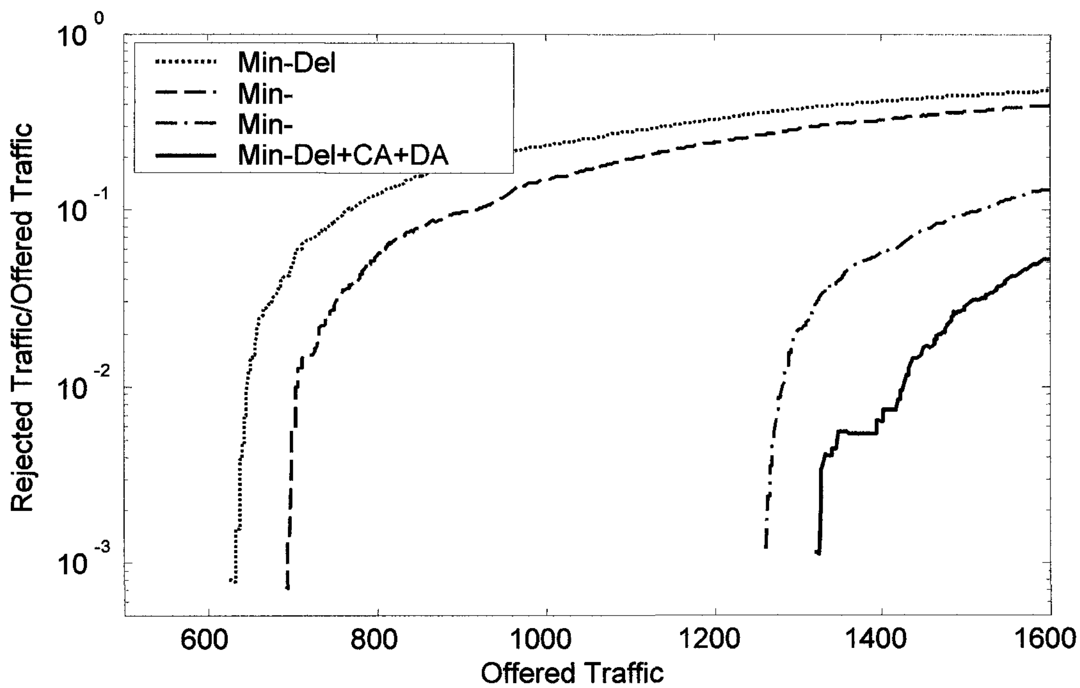


Figure 3-10: Class and delay assignment applied to Min-delay routing

presents results for both the centralized and the online approximation. It is interesting to notice that when multi-class delay margin based routing configuration is used alone, it does not present a major enhancement over minimum delay, even when using the centralized offline version. However when combined with the other two algorithms it outperforms minimum delay routing combined with same algorithms by more than a 100 traffic units. Figure 3-11 shows that when combined with delay and class assignment the approximation provides reasonable performance in trade for its distributed nature. Compared with min-hop the advantage for using the delay margin approximation is that it is based on the same formulation as the delay and class assignment. This enhances the interaction between these algorithms as they all aim to achieve the same objective. The understanding of the nature of this interaction and conditions where each of the algorithms become more valuable are our current research focus.

3.8. Delay Margin route configuration using priority schedulers.

In priority queues the dependency between queues is much simpler as lower priority queues depend on higher priority queues. Because of this $L_{n,q,j}$ is calculated as:

$$L_{n,q,j} = \frac{1}{\gamma_m}\left[\frac{1}{\left(1-\hat{d}_{n,q,j} / \hat{D}_{n,q,j}\right)} + \sum_{e\in(E_{n,q})} l_{e,j}\right] \quad (3.16)$$

where $l_{e,j} = \sum_{v\geq j} \frac{\partial d_{e,i}}{\partial \lambda_{e,j}} \psi_{e,v}$, and

$$\psi_{e,v} = \sum_{m} \sum_{(p|e\in(E_{u,p,i}\cap E_{n,q,v}))} \frac{\gamma_{i,p,v}}{D_m\left(1-\hat{d}_{i,p,v} / \hat{D}_i\right)^2} \quad (3.17)$$

In the case priority queues are used delay operating point has to be on the edge of the delay feasible region and hence there is no need for the definition of the delay operating point. Only the *Route Configuration* and *the Route Traffic Promotion/Demotion* algorithms are used. In the priority queue scenario we stick to a more traditional DiffServ approach where promotion and demotion apply only to end-to-end flows. The model of a priority queue with Poisson packet arrivals [92] is used to estimate each class delay.

3.8.1. Simulation scenario

In the priority queue case we use both Network Net1 and Net2 and we use the same call admission as that used in 3.6.1. In the priority class case all generated trunks will have the same delay limit (QoS requirement) of 3.5 ms. Similar to Net 1, traffic generated by the calls is Poisson distributed, however, γ_c =2 for all calls in Net2. In the multi-class *dynamic* case, the call traffic is increased by 1.2 and 2 for Net1 and Net 2, respectively, to yield a high number of rejected calls for a better performance comparison of different approaches.

3.8.2. Priority scheduling static cases:

3.8.2.1. Effects of number of priority classes and class adjustment:

Figure 3-12 shows the unaccepted traffic ratio versus the network offered traffic for various MC-DRPM schemes. For the centralized (C) formulation, increasing the number of priority classes (I) decreases the unaccepted traffic ratio, especially for *V=1* to *V=2*. On the other hand, for the distributed implementation without end-to-end class adjustment (D), the performance with I=2 is worse than that with *V=1*. This is attributed to two reasons: first at low load the effect of other users on the cost function is minimum, hence converting the approach to a minimum delay routing approach. Secondly, the static nature of the experiment prevents calls from changing their previous route and priority assignments. Because minimum delay routing would send traffic over the highest priority class (min delay), calls arriving to the lightly loaded network will

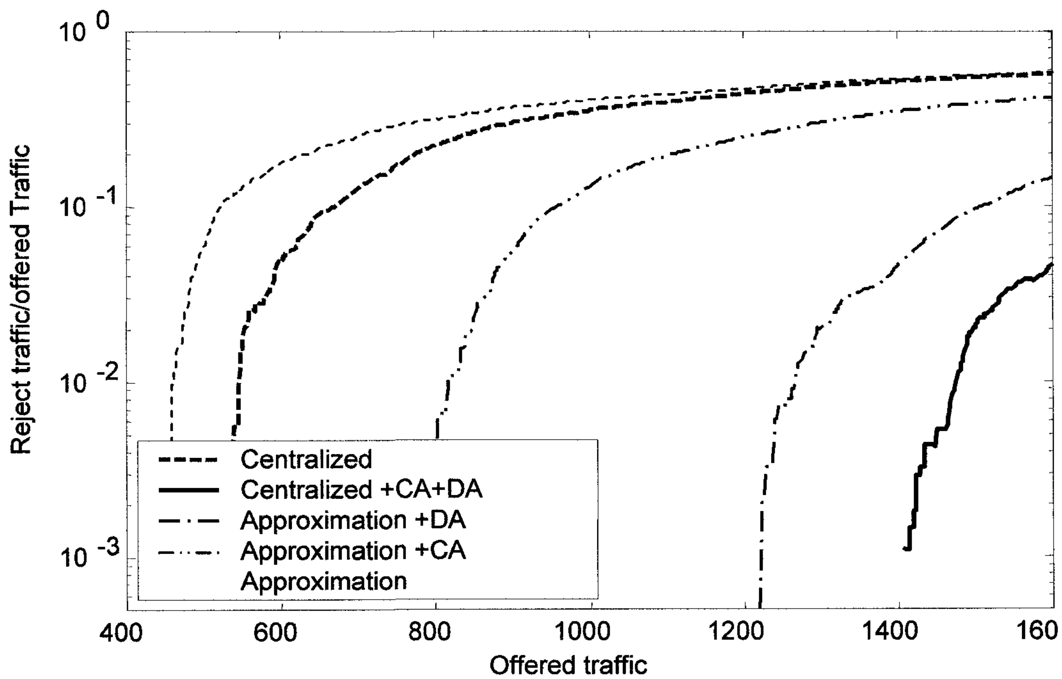


Figure 3-11: Delay Margin Route configuration centralized versus approximation

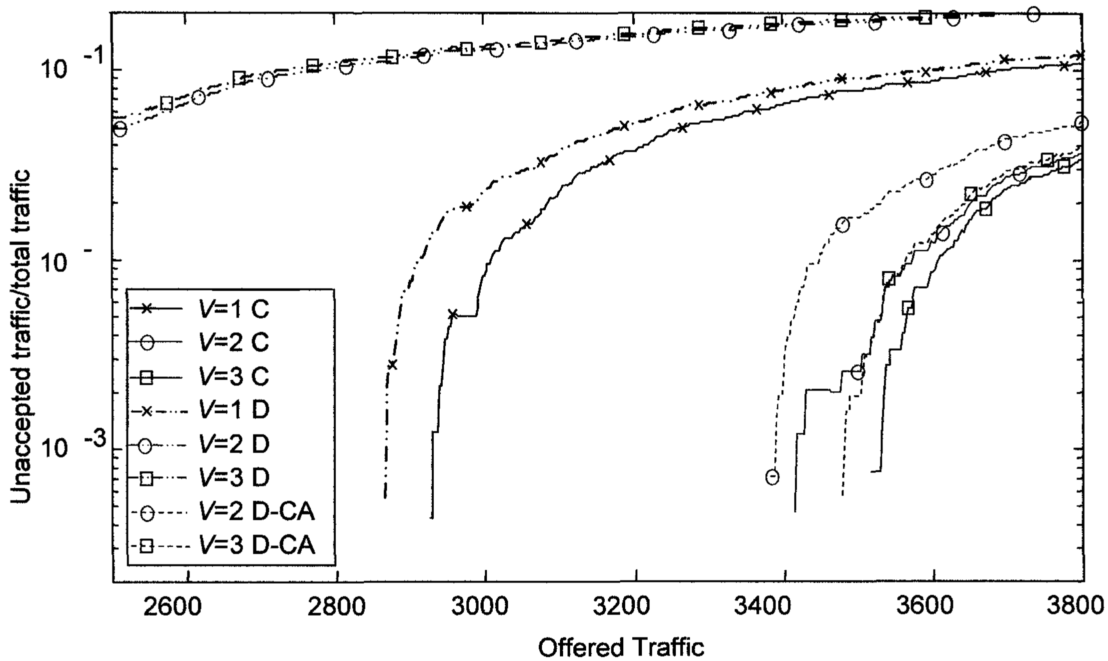


Figure 3-12: Centralized versus Distributed MC-DRPM for different numbers of supported priority levels: (Net1)

use the highest priority classes and will remain in it for the rest of the simulation. This causes an increase in the delay in both lower and higher classes and hence limits the capability of the network to accept more traffic. As discussed earlier, the class adjustment is distributed by nature as it can be done by each trunk independently based on the cost and delay prediction of other classes along the routes currently being used by the trunk. By including class adjustment, the performance of the distributed implementation with end-to-end class adjustment (D-CA) is closer to that of the centralized (C) formulation for V=2 and 3.

The same tendency is observed for Net 2 (Figure 3-13), i.e., increasing number of priority classes also improves the performance even in case of the same QoS (delay limit). This indicates the effectiveness of the MC-DRPM in appropriately assigning the right priority level for each flow on a given route. It is noted that a fixed priority assignment would have mapped all these trunks to the same priority level because they all have the same QoS requirement.

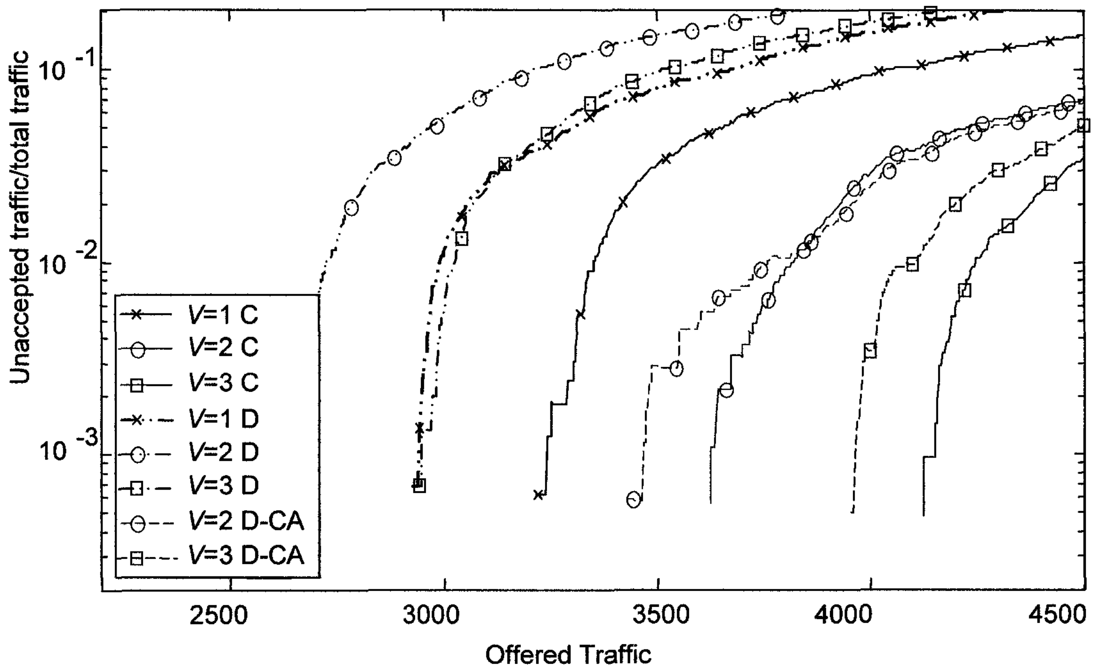


Figure 3-13: Centralized versus Distributed MC-DRPM for different numbers of supported priority levels: (Net2)

Compared to Figure 3-12, Figure 3-13 shows a larger improvement when the number of priority classes V, moves from 2 to 3 in Net2. Because Net2 has a larger number of trunks, the contention between trunks over links is bigger and hence the proposed algorithm becomes more effective in terms of mapping routes to priories in such a way that would decrease contention.

3.8.2.2. Effects of approximation in distributed MC-DRPM

Figure 3-14 and Figure 3-15 compare the performance (in terms of unaccepted traffic ratio) of the distributed MC-DRPM scheme. This is done using exact calculation of cost rate $L_{m,p,i}$ using Equ. (3.12) (D-CA-ER), and approximate cost rate calculation presented in section 3.5.1 (D-CA-AR). In both cases, the heuristic class adjustment step is employed periodically (every 50 simulation instances). The figures show that for V=1, there is a large difference between exact (ER) and approximate (AR) calculation. As the number of priority classes V, increases, the performance gets better and the difference between the exact and the approximations calculations is greatly reduced.

3.8.2.3. Distribution of traffic on priority classes:

To investigate the distribution of traffic over priority classes, we compare it with predefined priority assignment. We use a trunk predefined priority mapping based on mapping ranges of trunk QoS to a given priority class. Within each priority class, D-DRPM [50] is used to find routes based on the calculation of $L_{i,p,v}$ as defined by (3.16). The predefinition of priority allows the algorithm to treat each priority class as a separate network. However, it poses the problem of how to define the trunk to priority mapping. To find the optimal mapping, we define priority assign a trunk to class p if its required delay is in the range from priority-(p-1) delau to priority-p delay thresholds. The amount of accepted traffic out of 2,000 arriving calls for all the possible combinations of the priority-1 and the priority-2 thresholds (based on Net 1 delay requirements) is shown in Table 3-5. The last column in the table represents the two-priority

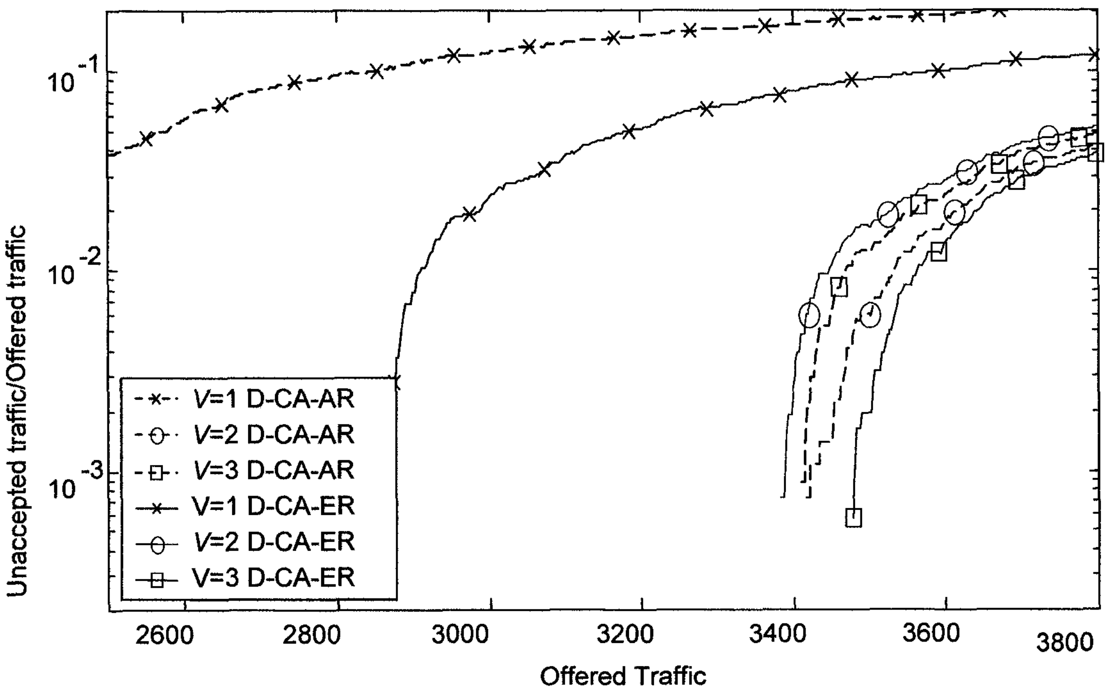


Figure 3-14: Exact versus Approximate calculation of cost rate $L_{m,p,i}$ (Net1)

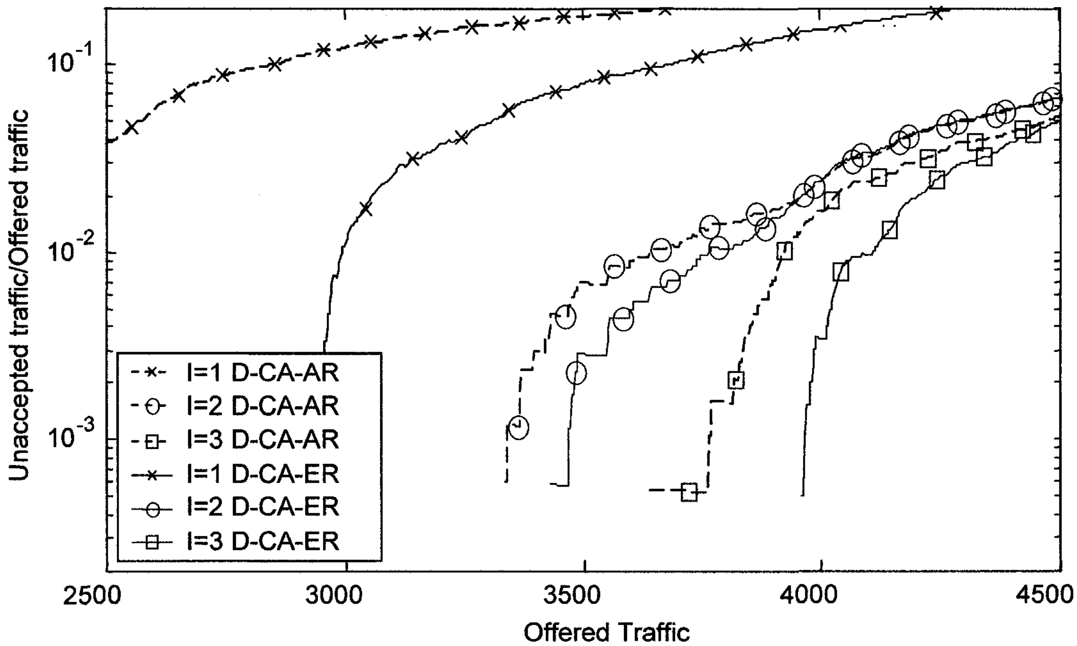


Figure 3-15: Exact versus Approximate calculation of cost rate $L_{m,p,i}$ (Net 2)

case. The results in the table show the total accepted traffic. The bold-faced number represents the maximum accepted user traffic. Figure 3-16 shows a performance comparison between the predefined priority assignments at the points where the total accepted traffic is maximum for both 2-priority and 3-priority network cases. Figure 3-16 shows that the centralized MC-DRPM (C) outperforms the predefined priority assignment (D-CA-PPA) and the performance of the D-MC-DRPM (D-CA) approaches that of the predefined priority assignment. The figure shows the ability of the D-MC-DRPM to approximate the optimal priority class assignment.

Table 3-6 shows the amount of trunk traffic accepted and assigned to each class for the 2-priority and 3-priority cases in the Net1 example, while Table 3-7 shows the same information for Net 2. For the case of 2 priority classes, Table 3-6 and Table 3-7 show that, the MC-DRPM provides a clear mapping from trunks to end-to-end priority levels. In the case of Table 3-6, that mapping is the same as the one obtained from Table 3-5. When three priority levels are used, the approach starts making use of the topology by assigning different trunk routes to different priority classes as shown by trunks 1, 2 ,4 in Table 3-6 and trunk 17 in Table 3-7. This allows the MC-DRPM to outperform the predefined priority assignment. It is also noticeable that the fixed assignment would have been inapplicable in the case of Net 2 since all the trunks have the same QoS requirement.

Table 3-5 : Total accepted traffic for different thresholds.

		Priority Class 2 threshold						
		.003	.004	.005	.006	.008	.01	0.12
Priority Class 1 threshold	.003	0	1507	3562	3641	3630	348	3360
	.004	0	0	3564	3647	3642	3551	3331
	.005	0	0	0	3676	**3667**	3676	3564
	.006	0	0	0	0	3635	3656	**3656**
	.008	0	0	0	0	0	3656	3614
	.010	0	0	0	0	0	0	3546

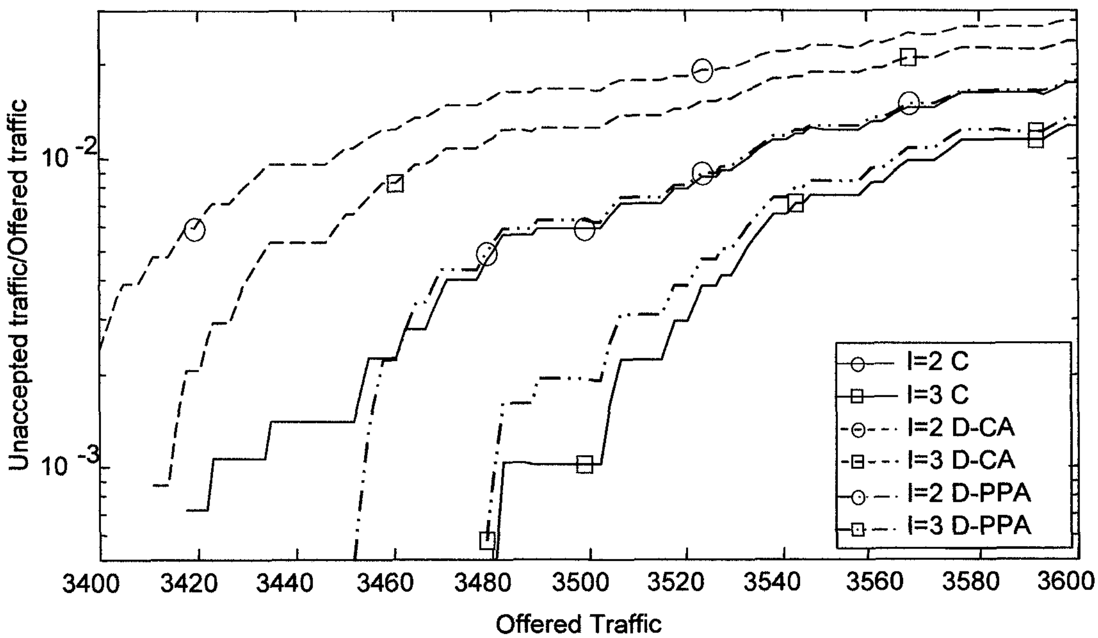


Figure 3-16: Performance of optimal pre-defined priority assignment (Net 1)

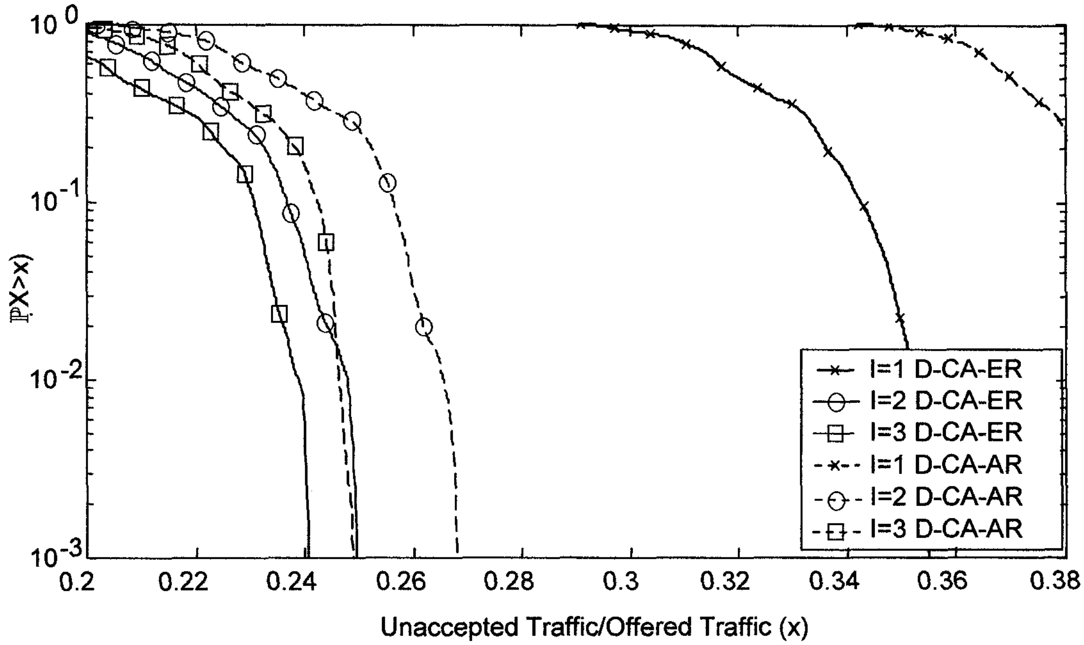


Figure 3-17: Performance in the dynamic case (Net1)

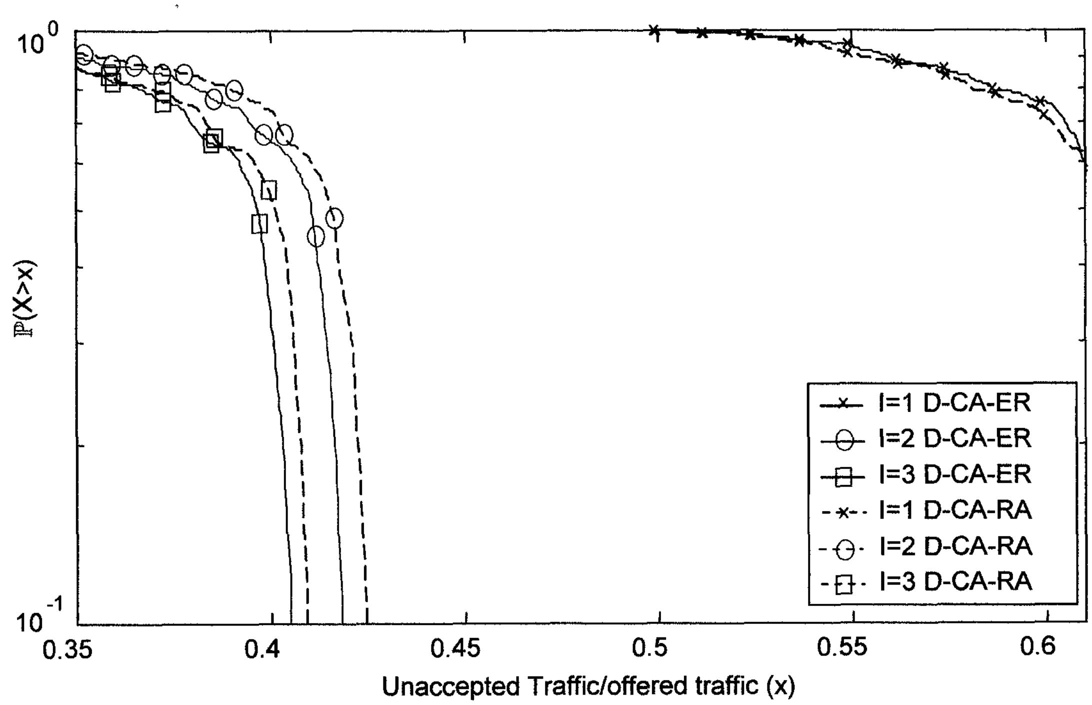


Figure 3-18: Performance in the dynamic case (Net2)

3.8.3. Priority scheduling dynamic cases

The dynamic case is mainly used to evaluate the distributed implementation of the algorithm and its associated approximation. We consider D-MC-DRPM with two and three priority classes. We used a simulation trace of 100,000 events, where each event signals the arrival time or departure time of a call in a given trunk. In all dynamic scenarios, each trunk performs adjustment of its priority classes every 10 time units. The probability that the rejection rate is larger than a certain amount for a given offered load is represented by the survival function in Figure 3-17 and Figure 3-18. The figures confirm that using multiple priority classes

Table 3-6: Accepted traffic per class per trunk using the centralized approach (Net1)

2 prio.	m	1	2	3	4	5	6	7	8
	i=1	525.7	2.325	653.8	442.4	0	0	0	0
	i=2	0	435.6	0	0	515.3	422.3	744.9	478.7
3 prio.	m	1	2	3	4	5	6	7	8
	i=1	24.16	105.5	0	299.2	0	0	0	0
	i=2	487.7	324.6	639.7	154.4	0	0	0	338
	i=3	0	0	0	0	515.2	428.4	720.1	0

Table 3-7: Accepted traffic per class per trunk using the centralized approach (Net2)

2 prio.	m	1	2	3	4	5	6	7	8	9	10	11
	i=1	168	0	0	0	198	0	0	0	0	0	178
	i=2	0	220	364	356	0	262	274	338	326	178	0
	m	12	13	14	15	16	17	18	19	20	21	
	i=1	136	0	172	0	0	164	0	0	194	0	
	i=2	0	346	0	324	286	0	332	186	0	256	
3 prio.	m	1	2	3	4	5	6	7	8	9	10	11
	i=1	168	0	0	0	198	0	0	0	0	0	0
	i=2	0	0	0	0	0	0	0	338	326	178	178
	i=3	0	220	364	356	0	262	274	0	0	0	0
	m	12	13	14	15	16	17	18	19	20	21	
	i=1	0	0	172	0	0	20	0	0	0	0	
	i=2	136	0	0	0	0	144	332	0	194	0	
	i=3	0	346	0	324	286	0	0	186	0	256	

outperforms the use of single class network with MC-DRPM. They also show that in the dynamic case, the approximate calculation of $L_{i,p,v}$ provides a comparable performance to the accurate calculation. This is mainly because the departure of users gives the network the chance to correct inappropriate assignments performed at earlier times due to the inaccuracy of the route length calculation.

3.9. Chapter Summary

In this thesis chapter, we presented a delay-margin based Traffic Engineering (TE) approach to provide end-to-end Quality of Service (QoS) in Multi-Protocol Label Switching (MPLS) networks using DiffServ at the link level. Three traffic engineering algorithms are developed using a nonlinear formulation of the TE problem in the form of end-to-end delay margin. The algorithms provide more control dimensions of the network, while keeping the simple DiffServ service provisioning architecture. Simulations show that the class assignment and the delay operating point assignment can be used separately to enhance the performance of existing routing techniques. In conjunction with the route configuration algorithm, they further increase the network performance. Approximations were also proposed for a possible distributed traffic engineering structure. When priority queues are used the approach is further simplified and results show that the proposed approach finds the optimal mapping between QoS requirements and priority levels.

Chapter 4

Analysis of GPS and Priority Queues with LRD Traffic Inputs and Variable Service Rates

This chapter provides the performance analysis for multi-queue systems that could be used for the delay margin based traffic engineering architecture discussed in Chapter 2. It focuses on the widely used weighted fair queues and priority queues. Using the techniques discussed in Chapter 3 for the proper choice of routing and delay class assignments clearly requires the ability of each link to characterize its current state as well as analyse its own performance. Each link needs to estimate the change of its delay and queue length distribution with the change of its load. In this thesis each link calculates $\partial d_{e,j} / \partial \lambda_{e,v}$. The calculation is done by analytically estimating the value of $d_{e,j}$ using the characterization of $\lambda_{e,v}$ that is measured at the current state and calculates $d_{e,j} + \Delta d_{e,j}$ at a perturbation of the traffic characteristics equal to $\lambda_{e,v} + \Delta \lambda_{e,v}$. The values of $\partial d_{e,j} / \partial \lambda_{e,v}$ can then be approximated by $\Delta d_{e,j} / \Delta \lambda_{e,v}$. The load change results from adding or removing end-user connections to/from a given link. Hence, each link needs to have the capability to estimate the change in the input traffic characteristics as a result of aggregating/segregating a give number of end-user applications.

Priority queuing can provide the service differentiation needed in the TE provided in Chapter 2, and it has the advantage of being simple and easy to implement which makes it a favorable technique for scheduling traffic at each output link in a DiffServ network. As indicated in Chapters 2 and 3 the functionality of the proposed TE can be further enhanced by using weighted fair queues. Weighted Fair Queuing (WFQ), the packet-based approximation of Generalized Processor Sharing (GPS) discipline [103], has been widely used for internet class-based QoS provisioning. While GPS weights guarantee a minimum capacity for each class, its work-conserving properties increase efficiency by redistributing the unused capacity between backlogged (active) queues. Accurate GPS performance estimation is important for understanding how GPS weights affect the QoS received by each queue.

The efficient resource utility offered by GPS comes at the price of increased difficulty in QoS performance estimation. The difficulty stems from the service rates dependencies between active and inactive queues. The use of GPS as part of a hierarchical work-conserving link-scheduling approach causes its service rate to vary. Variable service rates further increase the queue coupling because the remaining service rates of inactive queues are factors of the same link service variation. The efficient resource utility offered by GPS comes at the price of increased difficulty in QoS performance estimation due to the inter-dependency in the service rates between active and inactive queues. The use of GPS as part of a hierarchical work-conserving link-scheduling approach causes its service rate to vary. Variable service rates further increases the queue coupling because the remaining service rates of inactive queues are factors of the same link service variation.

When WFQs are used, the proper choices of delay operating points discussed in Chapter 3 requires the ability to estimate the effect of changing the weight of one class on its own performance as well as the performance of other classes. This chapter focuses on the performance analysis of priority queues and weight fair queues. To enable the analysis the chapter first presents analytical approaches that allows for using MSQ [95] to calculate the delay and queue length distributions for a single queue with variable service rate. The chapter then proceeds to evaluating the performance of priority queue then weighted fair queue.

For any queue in a priority queue scheduler, the evaluation is based on modeling the multi-queue as a two-queue system. In the two queue analytical model, all traffic of priority higher than the queue under consideration is aggregated and represented by one higher priority queue, while the queue under consideration is analyzed as the lower priority queue. To enable the analysis, the thesis first introduces a technique to estimate the multi-scale model for an aggregate of traffic sources in Section 4.3.1. The thesis then provides in Section 4.3.2 means for using Variable Service Multi-Scale Queuing VS-MSQ to obtain the multi-scale characteristics of the capacity that is not used by the higher priority traffic aggregate. Using this multi-scale model of the remaining capacity as the service rate to the lower priority queue the queue length and delay distributions are estimated using VS-MSQ and compared with simulation in Section 4.3.3. Comparison with simulation results show that the proposed analysis approach can provide a tight bound on the queue length and delay distributions.

Like priority queuing, the core of analyzing WFQ is in estimating the multi-scale service rate model for each of the queues. The estimation in Section 4.4 is based on a two dimensional decomposition. At first, temporal decomposition is used to convert the time-correlated queuing problem into a set of sub-problems over several time scales. Subsequently, queue decomposition exploits the queue weight dependencies to convert the multi-queue problem into a set of single-queue problems. The thesis shows the hierarchy of this estimation and the dependency of the queue service rate on the unused capacity of other queues and their weights. Section 4.5 provides comparison between simulation and analytical results

4.1. Related Work

Various analytical techniques to investigate the performance of priority queuing systems were presented for short-range dependent (SRD) traffic (e.g., [96]-[101]). Markovian characterization of the queue output process, based on exponential modeling of the periods of zero queue length in the high priority queue was used in [101]. Characterization of the queue output process is used to find the service rate for the lower priority queue. The approach in [101] is based on assumptions that are more valid for Markovian and SRD traffic modeling. Simulations in [69] show that delay and queue length characteristics are greatly affected by the traffic self-similarity. In the existence of self-similar traffic behavior, the use of Markovian approximations would render the analysis inaccurate.

Empty Buffer approximation and Reduced Service Rate [102] were proposed to deal with SRD or LRD traffic. Empty Buffer Approximation [102] assumes that the amount of high priority queue traffic is very small compared with the low priority traffic. This causes the high priority queue departure process to be the same as the input traffic process. In this case the queue length distribution of the lower priority queue approximates the queue behavior of a FCFS queue with an input traffic of both the high and low priority traffic. Reduced Service Rate [102] models service rate for the low priority class with a reduced rate where the reduction is equal to the long-term average usage of the higher priority queue. Both approximations are valid only for certain mixes of high priority and low priority traffic. These approximations are based on transforming the priority queue problem to FCFS queuing problem with constant input rate. Such an approximation is not accurate for higher volumes of high priority traffic or if this traffic displays high degrees of variation or self-similarity. There is no analytical technique capable of accurately capturing the effect of the traffic LRD or self-similar behavior. Such an analytical technique is required for QoS analysis and for the support of both the route configuration and the dynamic end-to-end QoS correction.

The queuing analysis of a Weighted Fair Queue or a Priority Queue GPS scheme is often started with queue decoupling to characterize the available service rate for each queue. This queue decoupling involves two main parts: (i) modeling the process of each *active* queue for all different combinations of active/inactive queues and, (ii) establishing the distribution of the remaining services (from *inactive* queues) to the active queues.

Three main approaches were used to characterize the GPS queue decoupling: Ordering, load-equivalence and Markovian. Ordering approaches focus only on the worst-case scenario. The GPS feasible ordering [103] effectively provides a priority-like queuing system where each queue uses a weighted share of the capacity remaining from its preceding queues in the order.

Using equivalent bandwidth, partial feasible ordering [104] establishes a statistical queue length bound. Because several feasible orders could exist, bounds obtained using [103] are loose. Grouping queues that share similar dependency on other queues into feasible partitions [104] establishes a unique order that causes the obtained bound to become tighter. Partial feasible ordering and feasible partitioning were used to derive queue length bounds for input traffic with exponentially bounded burstiness [104]-[105], Weibull-bounded burstiness (WBB) [106], Fractal Leaky Bucket Policing [107], and for a Markovian ON/OFF source [108].

Due to the worst-case assumption, ordering approaches provide pessimistic performance estimation, especially for Long-Range Dependent (LRD) traffic inputs. Although [106] attempted to capture the effects of LRD by including the traffic bounding process tail in the ordering, the conditions presented in [106] do not always guarantee a feasible ordering. Because [106] and [104] are based on the concept of equivalent bandwidth which considers traffic at a specific time scale, the resulting ordering cannot capture the inter-queue temporal correlation introduced by LRD traffic inputs at other time scales.

Reduced load equivalence approaches [109] aim to find the average service rate seen by the queue. They can be used as in [110] to obtain a bound on the exponent of the queue-length survivor function, when the average arrival rate of an aggregate of ON/OFF LRD sources is smaller than their weighted shares of the link service rate. Otherwise induced burstiness [111] may occur. Despite the simplicity of the results in [110] and [111], they lack the detailed explanation of the temporal correlation between different flows and the ability to consider the traffic inputs fine variation.

Both the ordering and reduced load equivalence approaches use the coarse characterization of the traffic, e.g., traffic average rates and, equivalent bandwidth. Markovian fine characterization of the decoupled service rate variation with time in a 2-queue GPS was used in [108] to find lower and upper bounds on the queue length distribution. For each queue, the formulation presented in [101] is used to derive the queue length upper bounds based on the assumption of exponentially distributed active queue periods, and a lower bound by assuming that the other queue is buffer-less. To overcome the increase in the Markovian state space in the multi-queue GPS case, the upper bound in [108] assumes that only one queue is inactive at a time. A more accurate Markovian solution that is presented in [112], considers the different combinations of active and inactive queues. When considering LRD traffic, the presence of variable service rates would complicate the Markovian solution by increasing the state space and would render the Markovian exponential modeling of each queue busy period inaccurate.

This chapter presents an analytical technique to estimate the queue length and delay distributions for Multi-Queue Systems using generalized processor sharing or priority queuing scheduling with LRD traffic inputs and variable service rates based on a *two-dimensional*, *multi-level* decoupling approach. One dimension handles the *temporal* correlation while the other deals with *inter-queue* coupling. The temporal decomposition uses multi-scale characterization of the traffic. For each time scale, a random process characterizes the total traffic arrival and total available service during the time interval defined by this scale. At each scale, the queuing problem is simplified to the difference between the total number of bits arriving and the total bits served during the time interval defined by this scale. Packet accumulation in the queue from previous time intervals is characterized using the total arrival and service over a longer time scale. The variation of queue distribution at successive scales captures the effect of the traffic and service temporal correlation on the queue length and delay distribution. A similar concept of analyzing the GPS queue length based on Gaussian characterization of the arrivals over different time intervals was presented in [113]. Unlike the analysis in this chapter, where the goal is to exploit the effect of all time scales on the queuing behavior, the analysis in [113] obtains an approximation of the queue length in a time scale corresponding to the queue length distribution most probable path. In this thesis, considering all time scales makes the estimation more accurate specially when dealing with LRD traffic where all time scales contribute to the queue performance.

The inter-queue decoupling focuses on accurately defining the service rate available to each queue by exploiting the dependency of each queue service rate on the activity and service rates of other queues. At each decoupling level, the service rate exceeding the queue traffic arrival (remaining service) is added to GPS minimum service rate assigned to other queues. The resulting service rate forms a new GPS queuing problem that is only limited to queues that do not have any remaining service rate. The resulting service rate creates a new set of active and inactive queues and a new level of decomposition is then performed that produces another new set of active and inactive queues. More levels of inter-queue decoupling are exploited until the remaining capacity is not usable by the queue under consideration. Unlike the feasible ordering/partitioning, the proposed decoupling approach in this thesis considers the remaining service rates from all queues in the system without assuming any order. By considering this remaining service rate over multiple time intervals, the proposed technique captures both the inter-queue coupling and the queue length temporal correlation.

Although the proposed multilevel inter-queue decoupling approach is applicable to various traffic models, the use of multi-scale traffic and service rate modeling provides simpler yet more accurate analysis of each queue output process. For this, we adopt the Multi-Scale Queuing (MSQ) model presented in [114] and its Beta-distributed multipliers to approximate the characterization of LRD traffic and variable service rates in the analysis. In [114], the MSQ model was used for the performance evaluation of single First-Come-First-Served (FCFS) queues with fixed service rates and LRD traffic inputs. The analysis in [114] did not have a provision for variable service rates, multi-queue systems, and delay consideration. As an extension, in [52], we developed an MSQ model for Variable Service rates (VS-MSQ) to analyze priority queues. Using VS-MSQ to analyze the decoupled queues eliminates the need to consider large state space and to assume certain Markovian characteristics of the queue busy period. It also simplifies delay distribution[3] analysis by examining the queue discharge over a time scale equal to the delay [53]. VS-MSQ was used earlier to analyze a two-queue GPS system with fixed service rate [54], where a much simpler queue coupling exists.

---

[3] It is far more challenging to obtain delay distribution using Markovian models.

4.2. Variable Service Multi-Scale Queuing (VS-MSQ) Analysis

4.2.1. General analysis:

4.2.1.1. Queue length:

Consider a queue with input instantaneous arrival rate $k\ (t)$ and service rate $c(t)$ observed in the time scales, $[-\tau_m, 0]$ for m=0,1,2,..,M where $\tau_m > \tau_{m-1} > 0$. The queue length Q at time t=0 can be represented as

$$Q = \sup\left\{q\left(\tau_m\right) + K[\tau_m] - C[\tau_m]\right\}, \tag{4.1}$$

where, $\sup\left\{a\right\} \triangleq \max\left\{a, 0\right\}$ the largest value among a and 0, $q(\tau_m)$ represents the queue length at time $-\tau_m$, and $K[\tau_m] = \int_{-\tau_m}^{0} k(t)dt$, and $C[\tau_m] = \int_{-\tau_m}^{0} c(t)dt$ represent the total traffic arrivals and the total amount of traffic serviced during $[-\tau_m, 0]$, respectively. Since the queue length is always non-negative, $q(\tau_m) \geq 0$,

$$Q \geq \sup_{m \in [0,1,\ldots,M]} \left\{K[\tau_m] - C[\tau_m]\right\}. \tag{4.2}$$

It follows that the probability that Q is smaller than a value q can be expressed as

$$\mathbb{P}(Q < q) \leq \mathbb{P}\left(\sup_{m \in [0,1,\ldots,M]} \left\{K[\tau_m] - C[\tau_m]\right\} < q\right), \tag{4.3}$$

where $\mathbb{P}(event)$ denotes the probability of an event, and

$$\begin{aligned}
&\mathbb{P}\left(\sup_{m \in [0,1,\ldots,M]} \left\{K[\tau_m] - C[\tau_m]\right\} < q\right) = \mathbb{P}\left(K[\tau_m] - C[\tau_m] < q, \forall m = 1,2,..,M\right) \\
&= \mathbb{P}\left(K[\tau_M] - C[\tau_M] < q\right) \prod_{m=M-1}^{0} \mathbb{P}\left(K[\tau_m] - C[\tau_m] < q \middle| K[\tau_{m+i}] - C[\tau_{m+i}] < q, \forall i = 1,\ldots,M-m\right)
\end{aligned} \tag{4.4}$$

In Equation (4.4) each of the events $(K[\tau_m] - C[\tau_m]) < q$ at different values of m is correlated with all events at time scales smaller than m (i.e from time scale 1 to $M - m$). The correlation stems from the fact that $K[\tau_m] \geq K[\tau_{m-1}]$ and that $C[\tau_m] \geq C[\tau_{m-1}]$ because arrivals within [-τ_m, 0] are equal to arrivals within [-τ_{m-1}, 0] plus the added arrivals within [-τ_m,-τ_{m-1}]. The same applies to the service rates at $C[\tau_m]$ and $C[\tau_{m-1}]$. Measurements in [57] show that the only significant autocorrelation in a multi-scale model exists between successive time scales. However, it can be seen that as the value of τ_m moves closer to the value of τ_{m+1}, the amount of arrivals and service within [-τ_m,-τ_{m-1}] becomes very small, and $\mathbb{P}\left(K[\tau_m] - C[\tau_m] < q \middle| K[\tau_{m+1}] - C[\tau_{m+1}] < q\right) \approx 1$. On the other hand, as τ_m becomes sufficiently smaller than τ_{m+1}, the amount of arrivals/service within [-τ_m,-τ_{m-1}] play a bigger role in determining the values of $K[\tau_m]$ and $C[\tau_m]$. This causes $K[\tau_m] - C[\tau_m] < q$ and, $K[\tau_{m+1}] - C[\tau_{m+1}] < q$ to become almost independent of each other, and hence

$$\mathbb{P}\left(K[\tau_m] - C[\tau_m] < q \middle| K[\tau_{m+1}] - C[\tau_{m+1}] < q\right) \approx \mathbb{P}\left(K[\tau_m] - C[\tau_m] < q\right).$$

This independence can be measured by the autocorrelation between the details in the two scales as explained in [57]. In this thesis these scale details are modeled by a Beta distributed multiplier as will be explained in Section 4.2.2. Using this autocorrelation the appropriate values can be chosen for τ_0 and for the maximum number of scales M. Furthermore, as $\tau_m \to \infty$ in a stable queuing system, then, $K[\tau_m \to \infty] < C[\tau_m \to \infty]$, and hence,

$$\mathbb{P}\left(K[\tau_m \to \infty] - C[\tau_m \to \infty] < q, q > 0\right) \to 1.$$

Based on the above discussion, we can adopt the following multiplicative multi-scale approximation model

$$\mathbb{P}(Q < q) \approx \prod_{m=0}^{\infty} \mathbb{P}(K[\tau_m] - C[\tau_m] < q) \approx \prod_{m=0}^{M} \mathbb{P}(K[\tau_m] - C[\tau_m] < q), \tag{4.5}$$

where, for good accuracy, the time-scales, $\tau_m > \tau_{m-1}$ for m=0,1,2,..,M, are selected to obtain

(i) $\mathbb{P}(K[\tau] - C[\tau] < q) \approx 1$ for $\tau < \tau_0$, (4.6)

(ii) $\mathbb{P}(K[\tau] - C[\tau] < q) \approx 1$ for $\tau > \tau_M$, and (4.7)

$$\text{(iii) } \mathbb{P}\left(K[\tau_m]-C[\tau_m]<q \middle| K[\tau_{m+1}]-C[\tau_{m+1}]<q\right) \approx \mathbb{P}\left(K[\tau_m]-C[\tau_m]<q\right) \quad , \quad \text{for } \forall m=0,1,\dots,(M\text{-}1) \tag{4.8}$$

For this and based on the evaluation in [118], and [114] we adopt the dyadic time-scale structure, i.e., $\tau_m = 2^m \Delta\tau$. Each term in (4.5) can be evaluated for independent $K[\tau_m]$ and $C[\tau_m]$ as

$$\mathbb{P}\left(K[\tau_m]-C[\tau_m]<q\right) = \int_{y=0}^{q} \int_{x=0}^{y} f_{K[\tau_m]}(y) f_{C[\tau_m]}(y+x) dx dy, \tag{4.9}$$

where $f_{K[\tau]}(y), f_{C[\tau]}(x)$ are the probability density functions (pdf's) of $K[\tau_m]$ and $C[\tau_m]$, respectively. The two main difficulties associated with this analysis are: (i) characterizing the pdf's of $K[\tau_m]$ and $C[\tau_m]$, and (ii) evaluating the integral in (4.9).

4.2.1.2. Delay:

The queuing delay d seen by a given packet arriving to a queue of length q, is the time needed to serve all packets that are in the queue at the point of the packet arrival. In other words, it is the time required to accumulate an amount of service equivalent to the queue length, q, i.e., $C[d]=q$ In the case of variable service rate $C[d]$ is a function of the service variation and its time correlation and hence the delay distribution is obtained as follows:

$$\mathbb{P}(D<d) = \mathbb{P}(Q-C[d]<0) = \int_{q=0}^{\infty} \int_{z=-\infty}^{0} f_Q(z+q) f_{C[d]}(q) dz dq = \int_{q=0}^{\infty} \mathbb{P}(Q<q) f_{C[d]}(q) dq \tag{4.10}$$

4.2.2. Traffic Characterization

The analysis in this thesis is applicable to any stochastic model that can provide an analytical form for the distributions of $K[\tau_m]$ and $C[\tau_m]$ at discrete values of τ_m. Discrete models that characterize the variation and the relation between the pdf's of $K[\tau_m]$ (at different values of m) are either additive (e.g.,[57]), or multiplicative (e.g., [119]). In this thesis, we use a multiplicative model that is similar to the one presented in [119]. We adopt the use of a multiplicative multi-scale model for its accurate simulation of LRD traffic (for comparison with analysis), and the flexibility of its Beta distribution. For dyadic time-scales (i.e., $\tau_m = 2^m \Delta\tau$) and for simplicity, $\Delta\tau$ is used as the time unit and is normalized to 1. $K[2^m]$ is modeled as the product of Beta-distributed multipliers using the following recursive relation

$$K\left[\tau_M\right] = A_M^{(K)} A_M^{(K)} \text{ , and } K[2^m] = K[2^{m+1}] A_m^{(K)} = A_M^{(K)} \prod_{j=m}^{M} A_j^{(K)} = A_M^{(K)} A_{*m}^{(K)} \text{ for } 0 \leq m < M \quad (4.11)$$

where 2^M is the coarsest time-scale. $A_m^{(K)} \in [0,1]$ is a multiplier for arrivals[4], which can be modeled as a Beta-distributed random variable [120] with the probability density function (pdf) $f_\beta(x; a_m, b_m)$ where

$$f_\beta(x;a,b) = \frac{x^{a-1}(1-x)^{b-1}}{\beta(a,b)}, \text{ and } \beta(a,b) = \int_0^1 x^{a-1}\left(1-x\right)^{b-1} dx \text{ is the Beta function.} \quad (4.12)$$

The values of $a_m^{(K)}, b_m^{(K)}$ are selected to match the 1st and 2nd moments of $A_m^{(K)}$. Since $\mathbb{E}(K[2^m]) = 0.5\mathbb{E}(K[2^{m+1}])$, $\mathbb{E}\left(A_m^{(K)}\right) = 0.5$ and hence $a_m^{(K)} = b_m^{(K)}$. Furthermore, its 2nd moment is $\mathbb{E}\left(\left[A_m^{(K)}\right]^2\right) = 0.5\left[a_m^{(K)}+1\right]\left[2a_m^{(K)}+1\right]^{-1}$. As the product of independent Beta random variables can be approximated by another Beta random variable [120], $A_{*m}^{(K)}$ is also Beta-distributed with the 1st and 2nd moments calculated as

$$\mathbb{E}\left(A_{*m}^{(K)}\right) = \prod_{j=m}^{M} \mathbb{E}\left(A_m^{(K)}\right) = 2^{m-M-1}, \quad \mathbb{E}\left(\left[A_{*m}^{(K)}\right]^2\right) = \prod_{j=m}^{M} \mathbb{E}\left(\left[A_m^{(K)}\right]^2\right) = 2^{m-M-1} \prod_{j=m}^{M} \frac{1+a_m^{(K)}}{1+2a_m^{(K)}},$$

and the pdf $f_\beta(x; a_{K[\tau_m]}, b_{K[\tau_m]})$ where

---

[4] referred to by the superscript (K)

$$a_{K[\tau_m]} = \frac{\mathbb{E}\left(A_{*m}^{(K)}\right)\left\{\mathbb{E}\left(A_{*m}^{(K)}\right) - \mathbb{E}\left(\left[A_{*m}^{(K)}\right]^2\right)\right\}}{\mathbb{E}\left(\left[A_{*m}^{(K)}\right]^2\right) - \left[\mathbb{E}\left(A_{*m}^{(K)}\right)\right]^2} = \frac{1 - \prod_{j=m}^{M}\left[1 + a_m^{(K)}\right]\left[1 + 2a_m^{(K)}\right]^{-1}}{\prod_{j=m}^{M} 2\left[1 + a_m^{(K)}\right]\left[1 + 2a_m^{(K)}\right]^{-1} - 1}, \text{ and}$$

$$b_{K[\tau_m]} = \left\{\left[\mathbb{E}\left(A_{*m}^{(K)}\right)\right]^{-1} - 1\right\} a_{K[\tau_m]} = \left\{2^{M-m+1} - 1\right\} a_{K[\tau_m]}.$$

$\Lambda_M^{(K)}$ is the maximum possible number of arrivals in 2^M. $A_M^{(K)}$ is the distribution of arrivals in 2^M normalized to $\Lambda^{(K)}$, where the parameters of the Beta-distribution modeling $A_M^{(K)}$ can be estimated by moment matching. M is chosen such that the correlation at $\tau_m > \tau_M$ will have negligible error to the queue estimation, e.g., $\mathbb{P}(Q[\tau_m] < q | Q[\tau_M] < q) \approx 1, \forall m > M$.

4.2.3. Queue Coupled Variable Service Multi-Scale Queuing (QC-VS-MSQ)

The multi-scale queuing (MSQ) analysis presented in [95] only deals with a fixed service rate, c. In this case, $C[\tau_m] = c\tau_m$ and (4.5) is evaluated as

$$\mathbb{P}(Q < q) \approx \prod_{m=M}^{0}\left[\mathbb{P}(K[\tau_m] < q + c\tau_m)\right] = \prod_{m=M}^{0}\int_0^{(q+c\tau_m)/\Lambda_M^{(K)}} f_\beta(x; a_{K[\tau_m]}, b_{K[\tau_m]})dx. \tag{4.13}$$

Equation (4.13) involves the evaluation of the terms of the product in the following descending order: M, (M-1),...,(m+1), m,...,1,0. The evaluation of the (m-1)$^{\text{th}}$ term needs the derivations of the Beta parameters by taking into consideration the resulting situation in the previous time scale m, $K[\tau_m] < q + C[\tau_m]$. For this we consider $\hat{K}[\tau_m] : K[\tau_m] < q + C[\tau_m]$ modeled as $\hat{K}[\tau_m] = \Lambda_M^{(K)}\hat{A}_{*m}^{(K)}$, where $\hat{A}_{*m}^{(K)}$ is a truncated Beta-distributed variable with pdf $f_{\beta'}(x; a_{K[\tau_m]}, b_{K[\tau_m]}, [q + C[\tau_m]] / \Lambda_M^{(K)})$ where $a_{K[\tau_m]}, b_{K[\tau_m]}$ are the parameters of the Beta-distributed variable $A_{*m}^{(K)}$

$$f_{\beta'}(x;a,b,c)=\begin{cases}\dfrac{x^{a-1}(1-x)^{b-1}}{\beta'(a,b,c)}, & \text{for } 0\leq x<c\\ 0, \text{ elsewhere}\end{cases},$$

where $\beta'(a,b,c)=\int_0^c x^{a-1}(1-x)^{b-1}dx$ is the incomplete Beta function.

Subsequently, in time scale (m-1), we assume $K[\tau_{m-1}]=\hat{K}[\tau_m]A_{m-1}^{(K)}=A_M^{(K)}\widehat{A}_{*m}^{(K)}A_{m-1}^{(K)}=A_M^{(K)}A_{*(m-1)}^{(K)}$ and $A_{m-1}^{(K)}$ is a Beta-distributed random variable with pdf $f_\beta(x;a_{m-1},b_{m-1})$. $A_{*m-1}^{(K)}$ is also a Beta-distributed random variable with *pdf* $f_\beta(x;a_{K[\tau_{m-1}]},b_{K[\tau_{m-1}]})$ and

$$a_{K[\tau_{m-1}]}=\frac{\mathbb{E}\left(A_{*m-1}^{(K)}\right)\left\{\mathbb{E}\left(A_{*m-1}^{(K)}\right)-\mathbb{E}\left(\left[A_{*m-1}^{(K)}\right]^2\right)\right\}}{\mathbb{E}\left(\left[A_{*m-1}^{(K)}\right]^2\right)-\left[\mathbb{E}\left(A_{*m-1}^{(K)}\right)\right]^2}, \text{ and } b_{K[\tau_{m-1}]}=\left\{\left[\mathbb{E}\left(A_{*m-1}^{(K)}\right)\right]^{-1}-1\right\}a_{K[\tau_{m-1}]}. \quad (4.14)$$

The 1st and 2nd moments of $A_{*(m-1)}^{(K)}$, are derived from the following moment matching:

$$\mathbb{E}(A_{*m-1}^{(K)})=\mathbb{E}(\widehat{A}_{*m}^{(K)})\mathbb{E}(A_{m-1}^{(K)}) \text{ and } \mathbb{E}\left(\left[A_{*m-1}^{(K)}\right]^2\right)=\mathbb{E}\left(\left[\widehat{A}_{*m}^{(K)}\right]^2\right)\mathbb{E}\left(\left[A_{m-1}^{(K)}\right]^2\right), \quad (4.15)$$

where $\mathbb{E}(A_{m-1}^{(K)})$, and $\mathbb{E}\left(\left[A_{m-1}^{(K)}\right]^2\right)$ are obtained from the statistics of the arrivals while the jth moment of $\widehat{A}_{*m}^{(K)}$ is calculates from $f_{\beta'}(x;a_{K[\tau_m]},b_{K[\tau_m]},[q+C[\tau_m]]/A_M^{(K)})$ as

$$\mathbb{E}\left(\left[\widehat{A}_{*m}^{(K)}\right]^j\right)=\int_{-\infty}^{+\infty}\mathbb{E}\left(\left[\widehat{A}_{*m}^{(K)}\right]^j\middle|C[\tau_m]\right)f_{C[\tau_m]}(x)dx, \quad (4.16)$$

where $f_{C[\tau_m]}(x)$ is the pdf of $C[\tau_m]$ and,

$$\mathbb{E}\left(\left[\widehat{A}_{m}^{(K)}\right]^{j}\left|C\left[\tau_{m}\right]\right.\right)=\frac{\beta'\left(a_{K[\tau_m]}+j,b_{K[\tau_m]},\left[q+C\left[\tau_m\right]\right]/A_M^{(K)}\right)}{\beta'\left(a_{K[\tau_m]},b_{K[\tau_m]},\left[q+C\left[\tau_m\right]\right]/A_M^{(K)}\right)}.$$

It is obvious from the above equation that for deterministic service rate, $\mathbb{E}\left(\left[\widehat{A}_{m}^{(K)}\right]^{j}\right)=\mathbb{E}\left(\left[\widehat{A}_{m}^{(K)}\right]^{j}\left|C\left[\tau_{m}\right]\right.\right)$. Furthermore, for fixed service rate of c, $C\left[\tau_m\right]$. For random service rate, we need to know $f_{C[\tau_m]}(x)$. For example, we can also model the variable service rate $C[2^m]$ using the following recursive relation

$$C\left[\tau_M\right]=A_M^{(C)}A_M^{(C)}, \text{ and } C\left[\tau_m\right]=C\left[\tau_{m+1}\right]A_m^{(C)}=A_M^{(C)}\prod_{j=m}^{M}A_j^{(C)}=A_M^{(C)}A_m^{(C)} \text{ for } 0\leq m<M \quad (4.17)$$

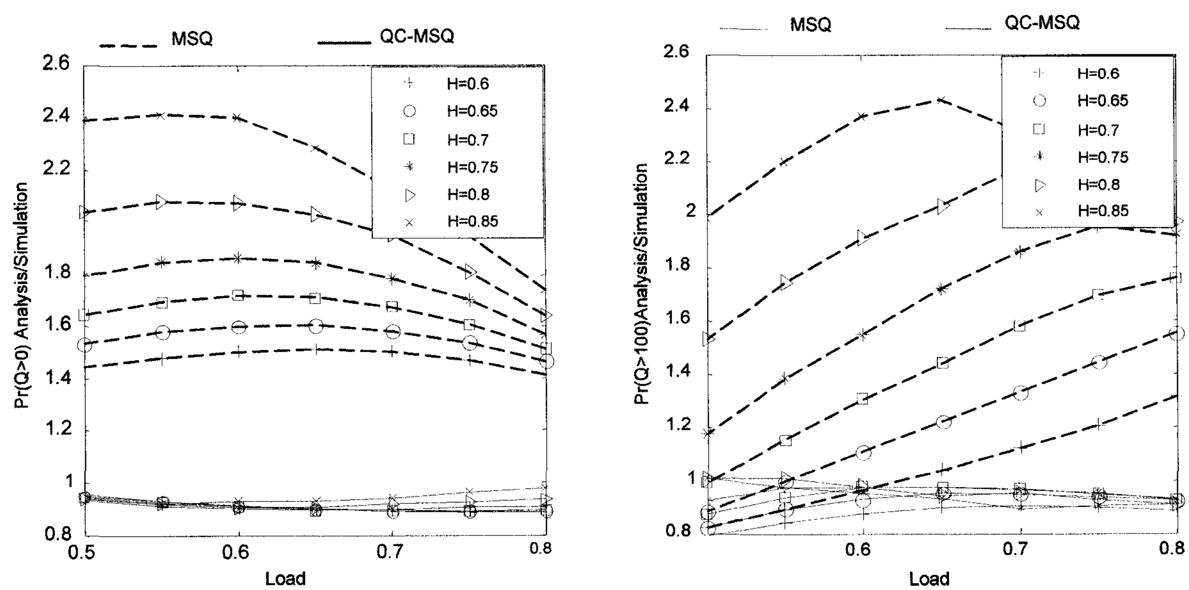
where $A_M^{(C)}$ is the maximum possible amount of available capacity in an interval 2^M, and $A_m^{(C)}\in[0,1], 0\leq m\leq M$ is a Beta-distributed multiplier for services, where its parameters can be estimated by moment matching.

4.2.4. Illustrative Results

This section compares the accuracy of the proposed *queue-coupled* VS-MSQ (QC-VS-MSQ) analysis described in sections 4.2.3 with the MSQ [95] and VS-MSQ [52] for both *fixed* and *variable* service rates. For reference, we consider the simulation results with the same multi-scale traffic model characterized by their mean μ, standard deviation σ, number of scales M, and Hurst parameter H (to characterize their level of self-similarity). The use of multi-scale in both the analysis and simulation makes sure that the difference between analysis and simulation is mainly due to inaccuracy in the analytical estimation, and is not caused by the multi-scale traffic model. For both the fixed and variable service rates, the same number of time scales M =15 are used. For the fixed service rate, a traffic trace with a mean μ =20 data units and a standard deviation σ =15 data units is used. Data units in this case can be bits, packets or cells. In the simulation, the load is varied from 0.5 to 0.8, and the Hurst parameter from 0.6 to 0.9. For the variable service rate, we use the same μ and σ of the traffic used in fixed service rate, but fix the Hurst parameter to 0.75. For the service rate the standard deviation is varied from 5 to 20 data units, the mean is set to 28.57 (0.7 load), and the Hurst parameter is varied from 0.55 to 0.9. The presented queue length results are also measured in data units. Each simulation point represents the average of 20 simulation runs with a 95% confidence interval.

4.2.4.1. Fixed Service Rate

Figure 4-2 shows the queue survivor functions obtained by the MSQ, QC-MSQ techniques, and simulation. While the *MSQ* overestimates the queue survivor function for short queue lengths, the proposed *QC-MSQ* provides an accurate estimation of the queue length for both long and short queue lengths (including zero queue length). As the queue length increases, both *MSQ* and *QC-MSQ* converge to the same value. This is mainly due to the fact that $\beta\left(a_{K[\tau_m]} + j, b_{K[\tau_m]}, (q + [\tau_m]c) / \Lambda^{(K)}\right)$ in (4.16) and approaches $\beta\left(a_{K[\tau_m]} + j, b_{K[\tau_m]}\right)$. This cancels the effect of the queue coupling when $(q + [\tau_m]c) / \Lambda^{(K)}$ falls in the region of the distribution tail where the probability of exceeding that value is very small. This also explains



(a) q=*0* (b) q=*100*

Figure 4-1: Comparison of queue survivor functions for fixed service rate

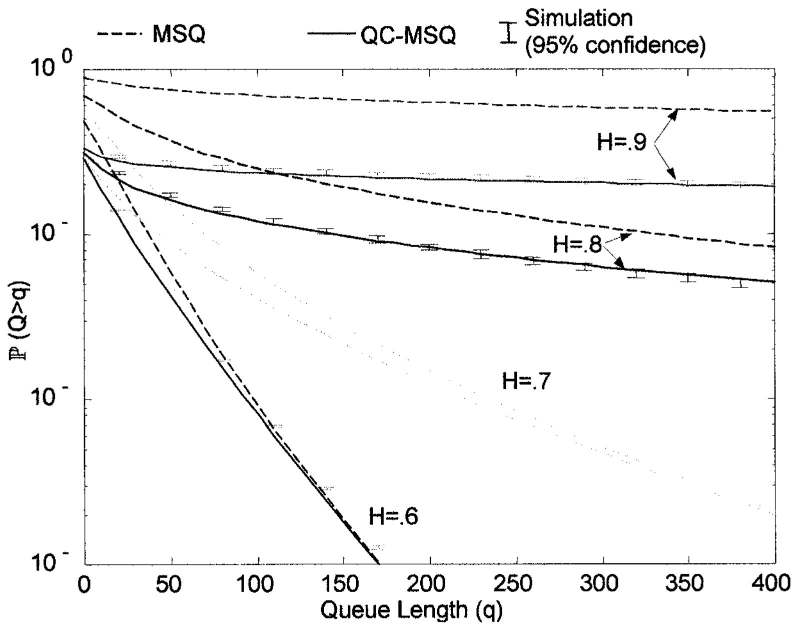


Figure 4-2: Queue survivor functions for fixed service rate (mean=20, standard deviation=16 and load= 0.65)

why the error increases as the Hurst parameter increases (and hence the self-similarity of the traffic). The increase in the Hurst parameter causes the variance of $K\left[\tau_m\right]$ at all scales to increase (for the same variance and mean at the lowest scale). The increase in the variance causes the tail of the distribution to decay at a slower rate and hence $\beta\left(a_{K[\tau_m]} + j, b_{K[\tau_m]}, (q + \left[\tau_m\right] c) / \Lambda^{(K)}\right)$ approaches $\beta\left(a_{K[\tau_m]} + j, b_{K[\tau_m]}\right)$ at larger value of q.

To have a better view of the accuracy, we use the simulation results as reference and examine the ratio of the queue survivor function given by *analysis* to that obtained by *simulation.* Figure 4-1 shows these ratios for both MSQ and QC-MSQ with $q=0$ and $q=100$. In general, MSQ overestimates the queue survivor function by more than double the simulation results and this over-estimation varies with the load and the queue length. On the other hand, the proposed QC-MSQ slightly underestimates the queue survivor function at low queue length and low Hurst parameter values. This inaccuracy can also be noticed in the MSQ at higher queue lengths and requires further investigation. The important part is that QC-MSQ estimation has very little variation with the change in Hurst parameters and the load.

4.2.4.2. Variable Service Rate

The queue survivor functions for a service rate with σ =10 plotted in Figure 4-3. Because, in the case of variable service rate, the queue delay is not exactly d / C, we also evaluate the delay survivor function by using (4.10) to obtain the analytical results for both VS-MSQ and QC-VS-MSQ shown in Figure 4-4. Simulation results shown in the figures are represented by their 95% confidence intervals. Both figures show smaller difference between VS-MSQ, QC-VS-MSC and simulation results. The proposed QC-VS-MSQ provides a more accurate estimation of the queue length and queue delay than the VS-MSQ.

For a better view of the accuracy, we use the simulation results as reference and examine the ratio of the queue survivor function given by *analysis* to that obtained by *simulation*. Figure 4-5 shows these ratios versus Hurst parameter for both QC-VS-MSQ and VS-MSQ and for two different values of σ . The accuracy of the MSQ is degraded as the Hurst parameter increases. At high Hurst parameters, the proposed QC-MSQ provides a much better accuracy than the MSQ.

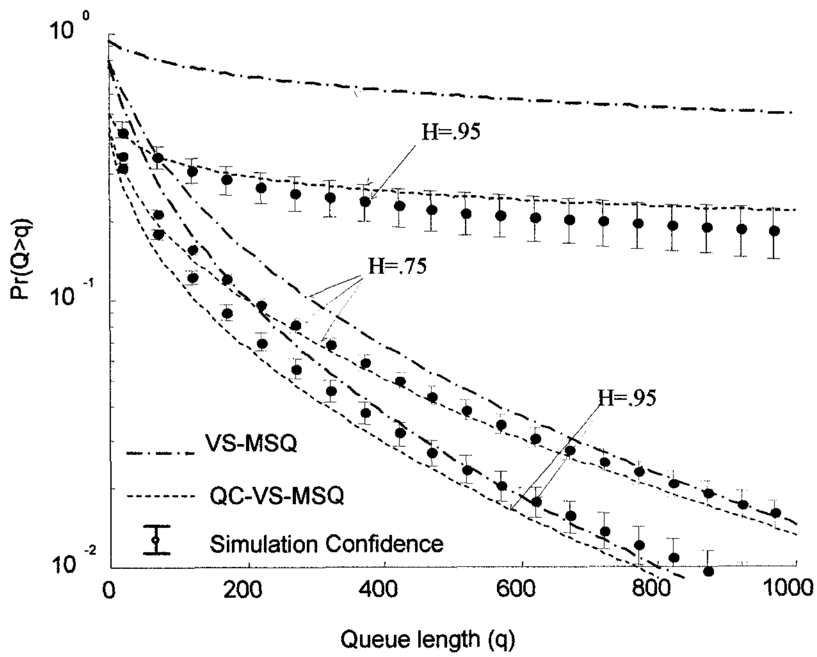


Figure 4-3: Queue survivor functions for variable service rate

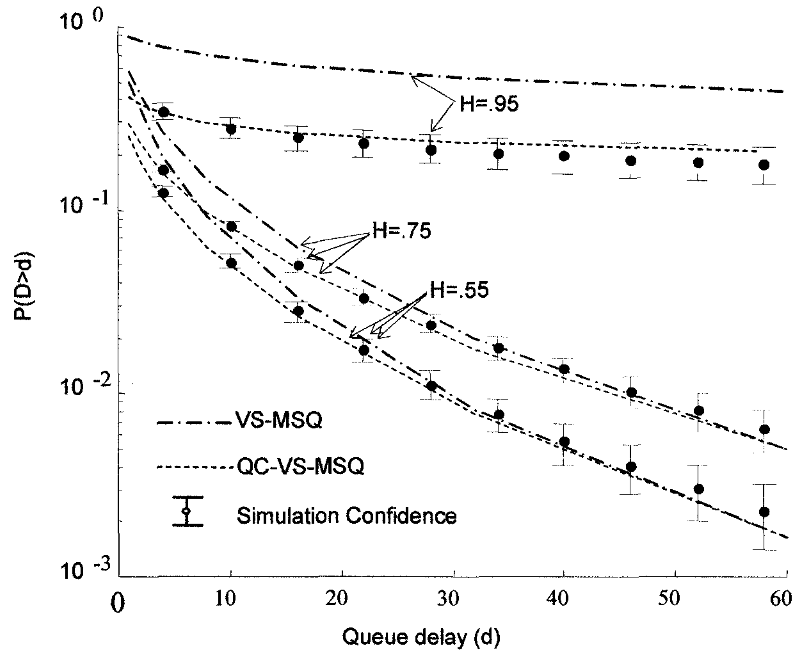


Figure 4-4: Delay survivor functions for variable service rate

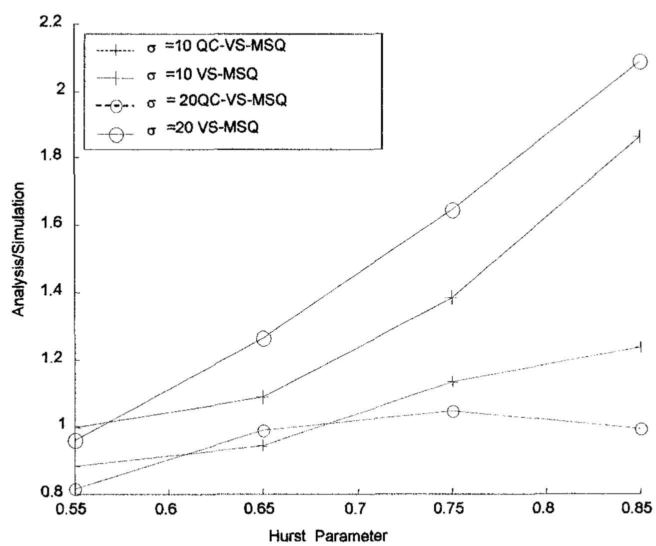


Figure 4-5: Comparison of queue survivor functions for variable service rate

4.3. Multi-Scale Priority Queue Analysis

The priority queue analysis in this thesis makes use of the simple fact that priority scheduling serves a queue of priority v with all the capacity remaining from the set of queues $\{1,2,.,.,v-1\}$ that have higher priority than queue v. This means that queue v sees a priority queue with V levels of priority as a two priority queuing system as shown in Figure 4-6. The higher priority queue in this two priority queuing system is fed with the aggregate traffic input to the higher $v-1$ queues and the lower priority queue is fed with the traffic input to queue v traffic. This reduces the analysis to four main steps. In the first step, stochastic models that characterize both the capacity available to the priority queue and the traffic arrival to each class are established. In order to enable the analysis in this thesis, both the service rate and the traffic arrivals are modeled using multi-scale models defined in Section 4.2.2. The second step deals with obtaining a stochastic model that characterizes the aggregate to the set of queues $\{1,2,.,.,v-1\}$. This aggregate will be used to characterize the input to the higher priority queue in the equivalent 2-queue priority queue system. At each time scale, m this aggregate traffic characteristic is evaluated as follows:

$$K_{1:v-1}\left[\tau_m\right]=\sum_{i=1}^{v-1}K_v\left[\tau_m\right] \tag{4.18}$$

In Section 4.3.1, this thesis provides a technique to find an approximation of the multi-scale model that characterizes an aggregate of traffic arrivals and evaluates the ability of the proposed approach to estimate the aggregate traffic characteristics for SRD and LRD traffic. The third step is focused on obtaining C_v the capacity that is available to queue v. In a priority queue C_v represents the capacity that was not used by the set of queues $\{1,2,.,.,v-1\}$. It is obtained by modeling a FCFS queue served with the total capacity C and has a traffic input equal the traffic aggregate obtained in the second. Using this model, this thesis proposes in section 4.3.2 a numerical technique to estimate the distribution of $C_v\left[\tau_m\right]$. The forth step uses multi-scale queuing to evaluate the queue length distribution of queue v using (4.5). The queue length distribution obtained using (4.5) can be used to evaluate the queue delay distribution using (4.10)

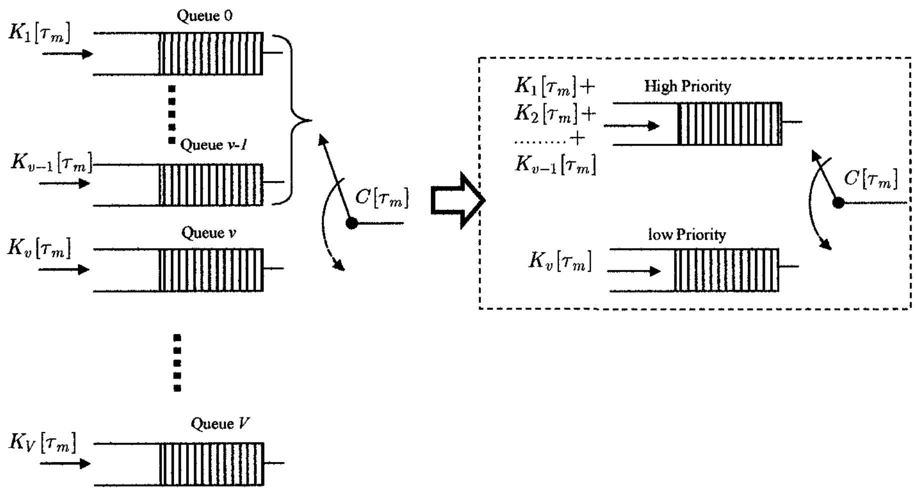


Figure 4-6: Analysis of queue v in a priority queue with V queues

4.3.1. Estimation of aggregate Higher Priority Aggregate Traffic

To estimate the parameters of the higher priority aggregate multi-scale model, this thesis uses a variance matching approach to estimate the parameters of a single trace.

Assuming that $A_{m,v}^{(K)}$ is the beta multiplier for traffic input to queue v at scale m . Let $\mathbb{E}\ (A_m^{(K)})$ and $\mathbb{E}^2(A_m^{(K)})$ represent the first and second moments of $A_{m,1:v-1}^{(K)}$ for the aggregate traffic arrival, where $\mathbb{E}\ (A_{m,v-1}^{(K)})$ and $\mathbb{E}^2(A_{m,v-1}^{(K)})$ can be evaluated as:

$$\mathbb{E}\ (A_{m,1:v-1}^{(K)}) = \left(\sum_{j=1}^{v-1} \mathbb{E}\ (A_{m,j}^{(K)}) A_j^{(K)}\right)\left(A_{1:v-1}^{(K)}\right)^{-1} \tag{4.19}$$

$$\mathbb{E}^2(A_{m,1:v-1}^{(K)}) = \left(\sum_{j=1}^{v-1}\left(\mathbb{E}^2(A_{m,j}^{(K)}) - \mathbb{E}\ (A_{m,j}^{(K)})^2\right)\left(A_j^{(K)}\right)^2\right)\left(A_{1:v-1}^{(K)}\right)^{-2} - \mathbb{E}\ (A_{m,1:v-1}^{(K)})^2, \tag{4.20}$$

where ,

$$A_{1:v-1}^{(K)} = \sum_{j=1}^{v-1} A_j^{(K)} \tag{4.21}$$

Using (4.19), (4.20) and (4.21) the multiplier $A_{m,1:v-1}^{(K)}$ can be fitted to a new symmetric beta distribution that is characterized by $a_{m,1:v-1}^{(K)}$, $m = \{1,2,....,M\}$ where

$$a_{m,1:v-1}^{(K)} = \frac{\mathbb{E}^2(A_{m,1:v-1}^{(K)})}{0.25 - \mathbb{E}^2(A_{m,1:v-1}^{(K)})} \tag{4.22}$$

This thesis uses two simulation scenarios to investigate the correctness of the proposed aggregate traffic characterization method. The first one considers LRD traffic and the other considers SRD traffic. The first scenario considers the aggregation of the Bell-core benchmark LRD traffic traces [115]. The second considers the characterization of an aggregate of SRD simulated voice sources. In both cases, beta parameters measured from the aggregate are compared versus those estimated. To further ensure the accuracy of the aggregate characterization, analytical queuing obtained using (4.5) for a FCFS queue fed with aggregate traffic that is characterized using (4.22) are compared versus those obtained using simulation.

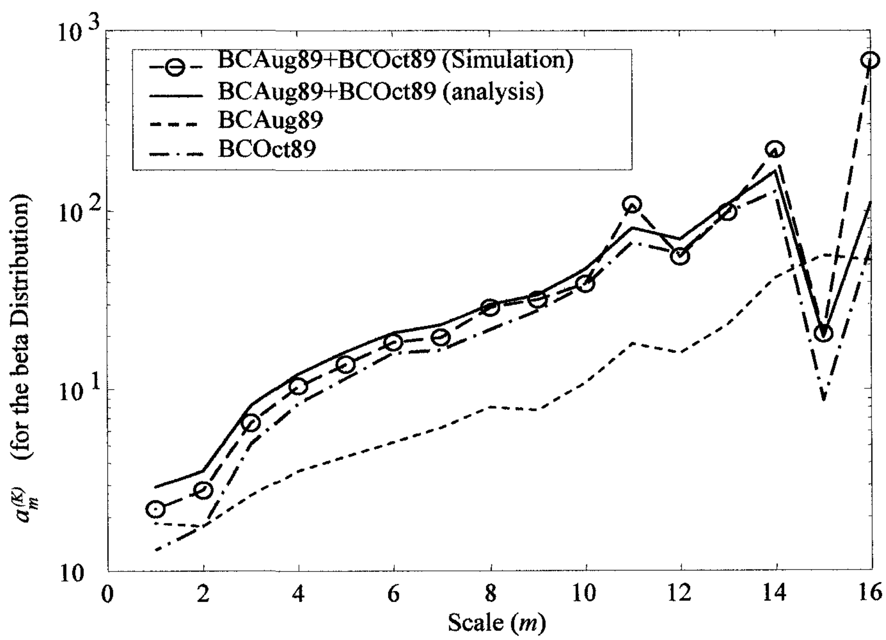


Figure 4-7: Beta parameters characterizing the aggregate of Bell-Core traces

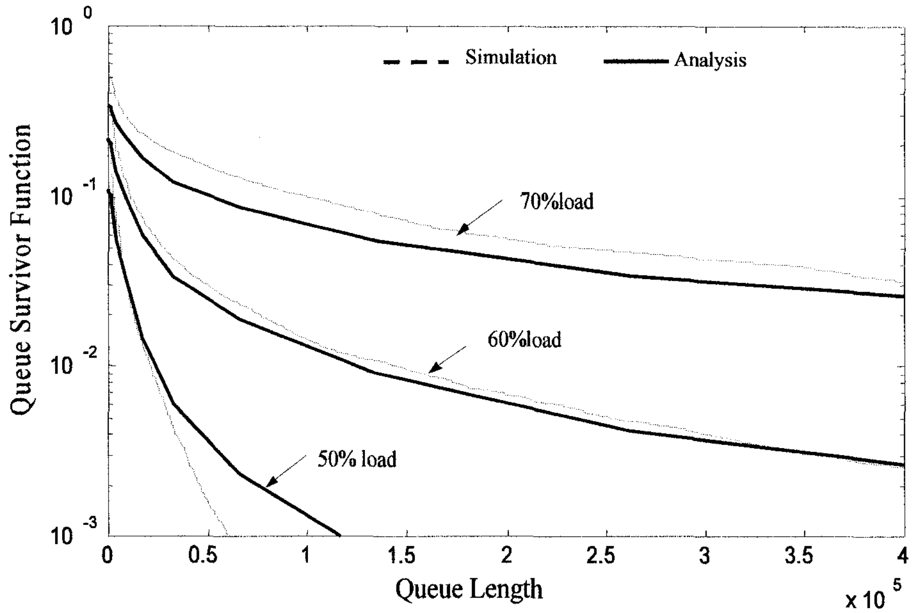


Figure 4-8: Queue Survivor function for the aggregate of Bell Core traces

4.3.1.1. Bell Core LRD traces

These traces were chosen because they represent actual traffic measurements and because they have been used as a benchmark in most studies about long-range dependent traffic. The thesis uses the Cell-core October 89 trace and the Bell-Core August 89 traces [115] which represent Ethernet chaptered packets during intervals of 3142.82 sec and 1759.62 sec respectively. In the evaluation, each trace is represented by the number of packets arriving during 10 ms intervals. The multi-scale parameters for both traces are measured, and are used to calculate the multi-scale parameters of the aggregate trace. Equation (4.5) is then used to calculate the survivor function of a FIFO queue fed with the aggregate traffic under 50%, 60%, and 70% loading using the calculated multi-scale parameters. The queuing results are compared with queuing simulation of these aggregate traffic arrivals over a FIFO queue

Figure 4-7 shows the comparison between the calculated and the measured multi-scale beta distribution parameters $a_m^{(K)}$ for the aggregate of the two *BC-Oct89* and *BC-Aug89* traffic. The figure shows that the technique proposed in section 4.3.1 is capable of estimating the beta parameters characterizing the aggregate correctly.

The comparison between the queue survivor function obtained using simulation and that obtain using analysis for different loads is shown in Figure 4-8. The comparison confirms that the aggregate traffic characterization proposed in section 4.3.1 can be used to accurately characterize the queuing behavior of the aggregate for different loading conditions.

4.3.1.2. Aggregate of Voice Sources

To investigate the ability of multi-fractal traffic models to model aggregates of SRD traffic, this thesis uses a simulation of the 12.2 kbps voice models defined by [116]. This model has an ON time that is exponentially distributed with an average of 3 sec, and an OFF period that is exponentially distributed with an average of 3 sec. Voice sources are used in this example to evaluate the ability of the technique proposed in this thesis to deal with traffic of practical value.

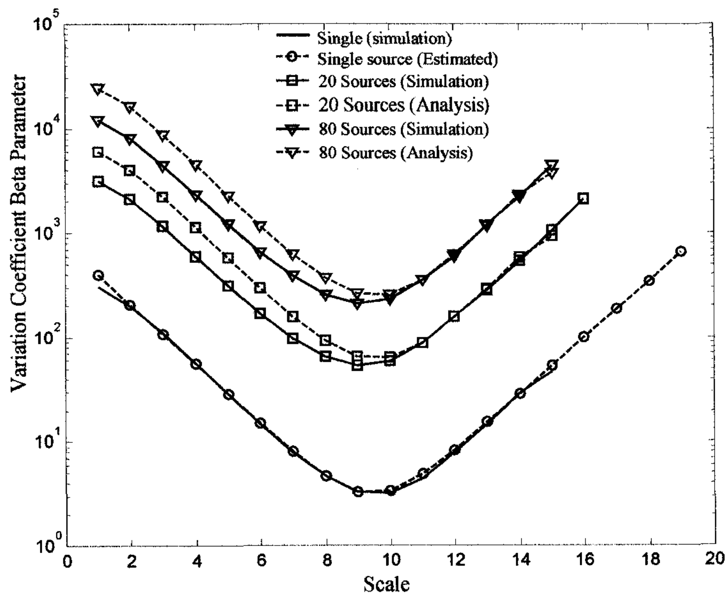


Figure 4-9: Comparison between the beta parameters for multi-scale model and measurements form simulation

The multi-scale parameters are measured for a single voice source and are used to estimate the multi-scale parameters for a varying number of sources. The results are compared versus a simulation of the same number of aggregates. Figure 4-9 shows that although the technique presented in section 4.3.1 captures the trend of the multi-scale function variation, however, the estimation accuracy decreases at lower time scales. This is mainly because the voice source has two states. The effect of each distribution is more obvious at lower scales. At longer time scales several ON and OFF periods can occur within the same time scale, hence one distribution (beta distribution can not be used to characterize the traffic). This is shown by the accuracy of the multi-scale traffic characterization at higher scale. Looking at the queue survivor functions in Figure 4-10 and Figure 4-11 for an aggregate of 20 and 80 sources respectively, it is seen that as the level of aggregation increases the queue estimation becomes more accurate. The voice source has two states, hence the pdf for a small number of sources has two peaks. Because the beta distribution can only approximate a *pdf* with one peak, the modeling accuracy decreases as the number of voice sources decrease. Using the central limit theorem, as the number of sources increase, the traffic distribution at all time scales becomes more bell shaped (Gaussian like). This enables the beta distribution to provide a more accurate approximation of the distribution.

4.3.2. Multi-Scale Characterization of Available Capacity

The service available to a lower priority queue v is the capacity that was not used by the higher priority queues 1 to $v-1$. The remaining capacity C_v is evaluated at scale τ_m as follows:

$$C_v\left[\tau_m\right]=\max(0,C\left[\tau_m\right]-K_{1:v-1}\left[\tau_m\right]) \tag{4.23}$$

Using (4.23), when the service rate available to queue v is bigger than zero, then its probability distribution function $f_{C_v[\tau_m]}$ can be evaluated as

$$f_{C_v[\tau_m]}\left(c\,|\,C_v\left[\tau_m\right]>0\right)=\left(1-\mathbb{P}\left(C_v\left[\tau_m\right]=0\right)\right)\int_{y=0}^{1-c} f_{K[\tau_m]}\left(y\right)f_{C[\tau_m]}\left(y+c\right)dy \tag{4.24}$$

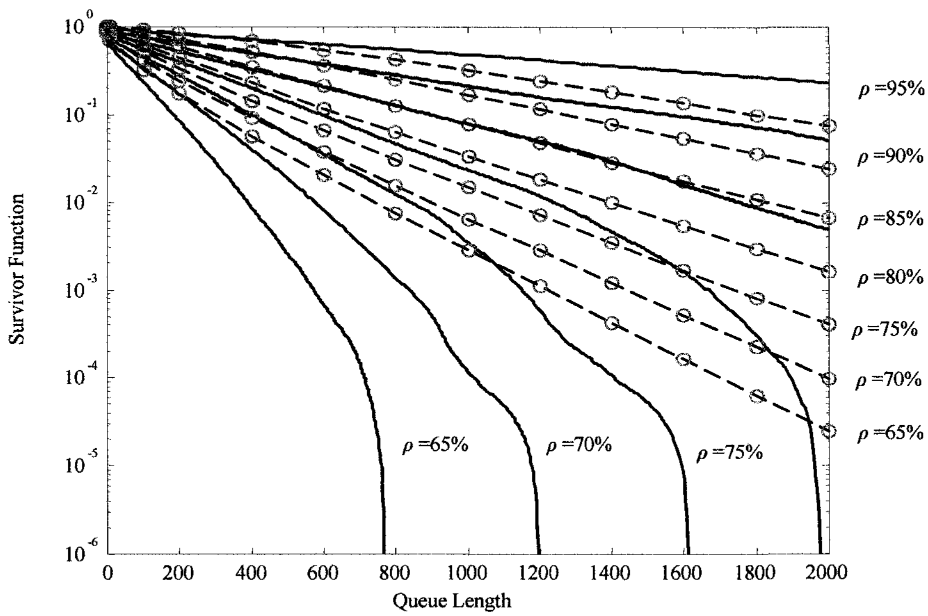


Figure 4-10: queue survivor function for 20 voice sources for different loads

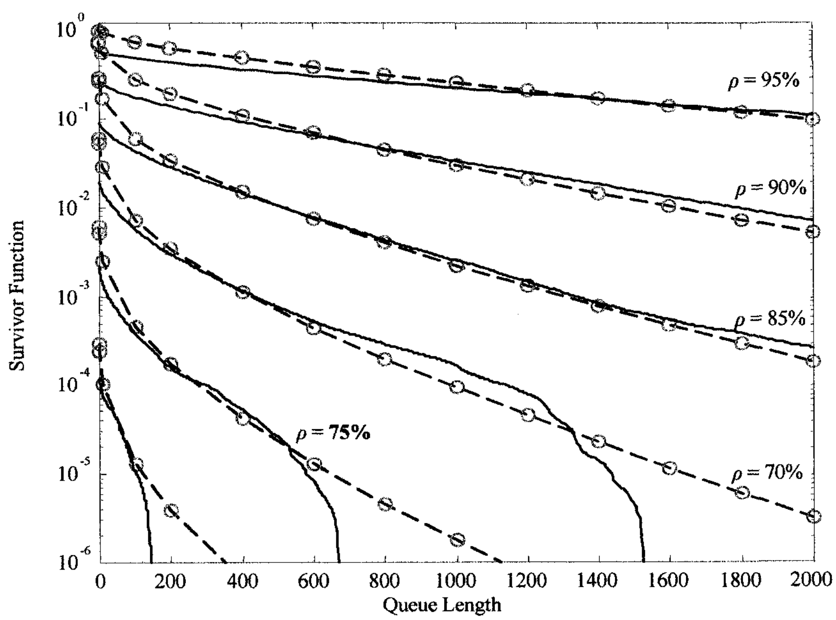


Figure 4-11: queue survivor function for 80 voice sources for different loads

Where $\mathbb{P}\left(C_v\left[\tau_m\right]=0\right)$ is the probability that the capacity available to queue v is equal to zero and can be evaluated as

$$\mathbb{P}\left(C_v\left[\tau_m\right]=0\right)=\int_{z=-\Lambda^{(K)}}^{0}\int_{y=0}^{\Lambda^{(K)}} f_{K\left[\tau_m\right]}(y) f_{C\left[\tau_m\right]}(y+z) dy dx \qquad (4.25)$$

The available service rate can not be equal to zero and hence

$f_{C_v\left[\tau_m\right]}(c)=0, c<0.$

Figure 4-12 and Figure 4-13 show a comparison of the available capacity distribution obtained using (4.24) and (4.25), with that measured over multiple time scales from queuing simulation of queues 2 and 3. In this example, the service rate of the priority queue is assumed to be variable with an average that is equal to the average input traffic divided by the load ρ, and the standard deviation σ of the capacity is adjusted such that it is equal to half the average service rate. The variation of the service rate can be the result of variation in the channel characteristic in the case of a wireless transmission or can represent the share of the queue when dynamic resource allocation is used. The service rate variation is assumed to be self-similar with Hurst parameter equal to H. The self-similarity used in this thesis represents an extreme case of service rate variation were temporal correlation in the service rate exists over multiple time scales. The modeling of the service rate variation in this thesis does not represent any real model but is chosen to test the ability of the proposed approach to estimate the queue length and delay distribution versus different loading conditions and different levels of self-similarity. The queue with the highest priority (queue1) in the analysis used in Figure 4-12 and Figure 4-13 is fed with an aggregate of 160 voice sources similar to the ones used in section 4.3.1.2 and queue 2 is fed with 32 video traffic sources. The analysis in Figure 4-12 and Figure 4-13 uses a scaled down version of the model defined in [117]. In [117] each video traffic flow is modeled by two ON/OFF processes. The ON model sojourn time is Pareto distributed to model the self-similarity of the traffic, where a cumulative distribution function of a Pareto distribution is represented by

$$\mathbb{P}(X<x)=\left(\frac{x}{a}\right)^{-\alpha}$$

Parameters for the video model used in this thesis section are listed in the Table 4-1

Table 4-1 Video traffic source parameters

	Packet size	Packet average inter-arrival	Pareto scaling a	ON period α	OFF period α	Average rate
Source 1	256	0.003559352	.04	1.14	1.22	38 Kb/sec
Source 2	256	0.0025873220	.04	1.54	1.28	42 Kb/sec

Similarly, queue 3 traffic is fed by 640 http traffic sources. Each http source is represented by an ON/OFF model. In the ON period, packets of size of 1240 bits are generated every 0.01 second. The ON sojourn time is modeled by a Weibull distribution with α = 0.7883, and a=0.64, where a cumulative distribution function of a Weibull distribution is represented by:

$$\mathbb{P}(X<x)=1-e^{-(x/a)^{\alpha}}$$

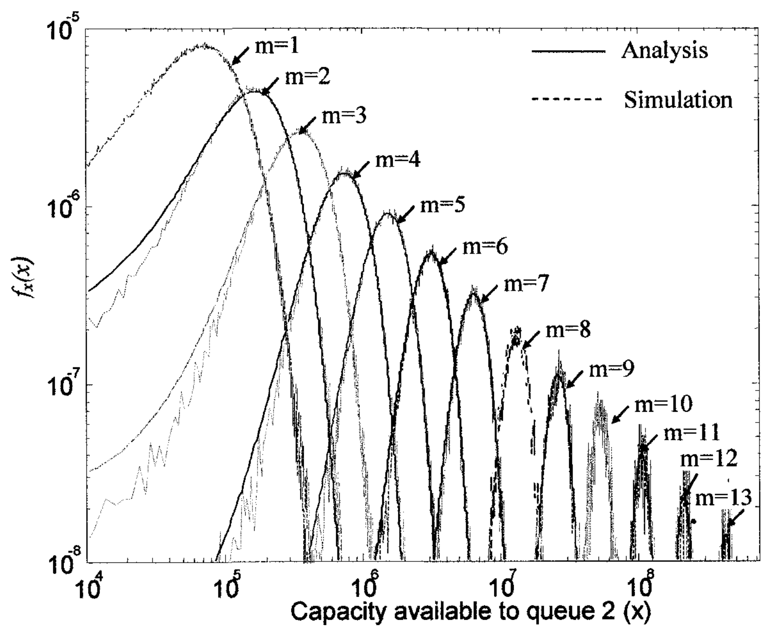


Figure 4-12: Example available capacity distribution for queue 2

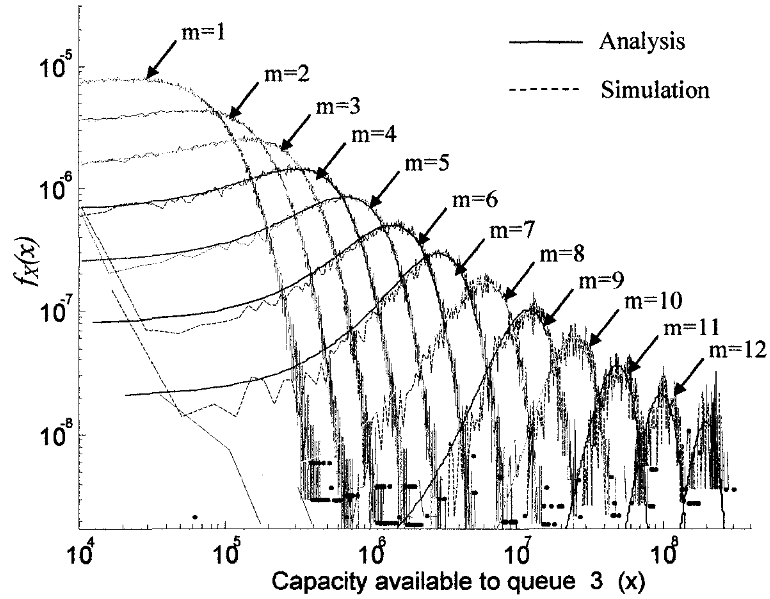


Figure 4-13: Example available capacity distribution for queue 3

The OFF sojourn time models the time between downloads and is represented by an exponential distribution with an average of 60 sec. The number of sources for each type of traffic is arbitrarily chosen to represent a priority queue load. The objective of the simulation is to evaluate the ability of the proposed technique to estimate the performance of the priority queue. The comparison in Figure 4-12 and Figure 4-13 shows that an accurate estimate of the capacity available to each lower priority queue can be obtained using equation (4.24) and equation (4.25)

4.3.3. Priority Queue Length and Delay Analysis

Using equation (4.5), each queue in the priority queue can be analyzed as a single FIFO queue that is being served with the multi-scale available capacity derived in section 4.3.2. The FIFO queue length and delay are both analyzed using the basic VS-MSQ technique presented in section 4.2.1 and the QC-VS-MSQ presented in section 4.2.3. QC-VS-MSQ is applied to the arrival using equation(4.16). To test the ability of the proposed approach in evaluating the performance of a priority queue, this section uses the same traffic and queue set-up that was used in section 4.3.2, but varies the service rate and the traffic self-similarity by varying the load ρ and the Hurst parameters H. Results are discussed in sections 4.3.3.1 and 4.3.3.2

4.3.3.1. Variation of self-similarity

Figure 4-14, to Figure 4-16 show the comparison between the calculated queue length survivor function (solid line) versus the survivor function obtained using simulation (dashed lines) at a loading of 70% and different levels of self-similarity. Each point on the simulation survivor function represents the average of 10 simulation runs. The 95% confidence interval levels are shown on the figure. Figure 4-14, to Figure 4-16 show that as the Hurst parameter of the capacity increases, the probability that the queue length exceeds a certain value increases. The figures shows that the approach presented in this thesis was able to provide an estimation for the effect of this variation. Figure 4-17 to Figure 4-19 shows that this queue length estimation can be used to provide a good estimation of the delay distribution for the 3 priority classes. Figure 4-14, to Figure 4-16 also show that as the self-similarity of the traffic increases the VS-MSQ analyses overestimated the queue survivor function. The over estimation is due to the increased role of inter-scale correlation. As the self-similarity of the traffic increased QC-VS-MSQ provided a better estimation of the queue length and delay distribution. At higher levels of self-similarity the queue induced inter-scale correlation increases and hence the role of inter-scale decoupling proposed in section 0 becomes more apparent.

4.3.3.2. Variation of load

When the self-similarity of the traffic is fixed at H=0.75 and the loading is changed from 50% to 90 % in Figure 4-20 to Figure 4-22 it is seen that the shift between the simulation and analysis increases with loading. Despite this shift in the survivor function estimation, the technique presented in this thesis have captured the increase in the queue survivor function with the increase in the load. The use of QC-VS-MSQ enables decreasing this shift for higher loads. At higher loads, the queue stays active for long periods, hence increasing the inter-scale dependency. The QC-VS-MSQ exploits this dependency to enhance the queue length estimation.

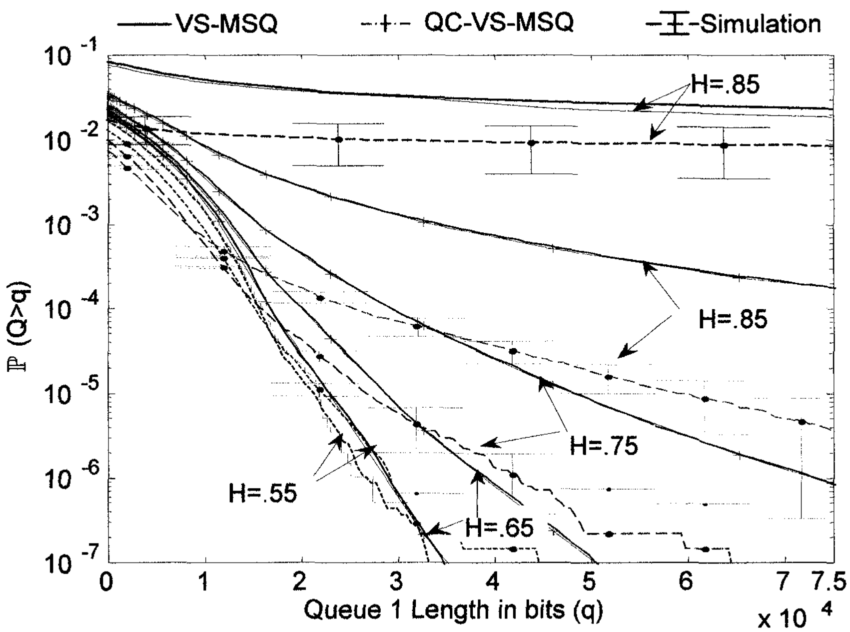


Figure 4-14: Queue length survivor function at a load of 70% for different service rate self-similarity for queue 1

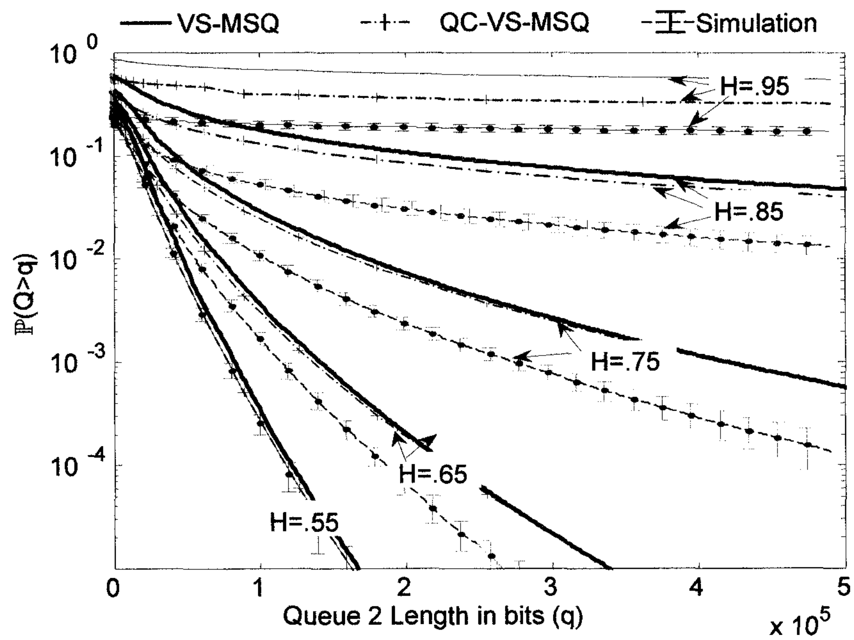


Figure 4-15: Queue length survivor function at a load of 70% for different service rate self-similarity for queue 2

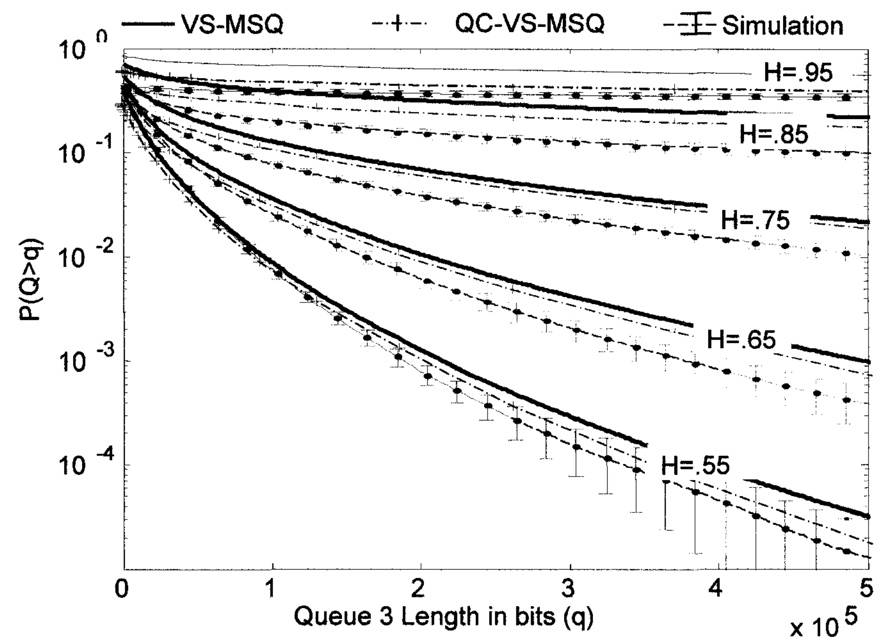


Figure 4-16: Queue length survivor function at a load of 70% for different service rate self-similarity queue 3

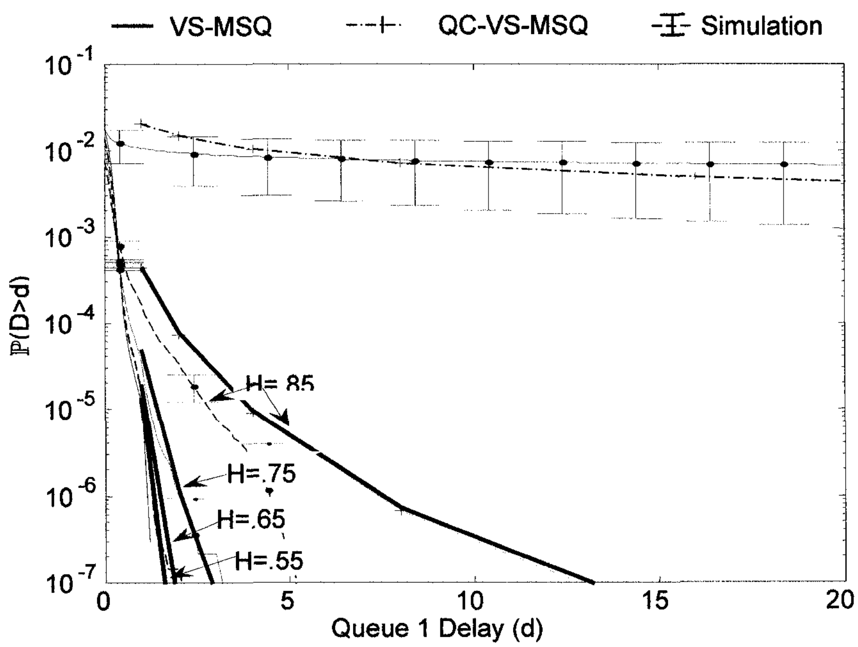


Figure 4-17: Queuing delay survivor function at a load of 70% for different service rate self-similarity for queue 1

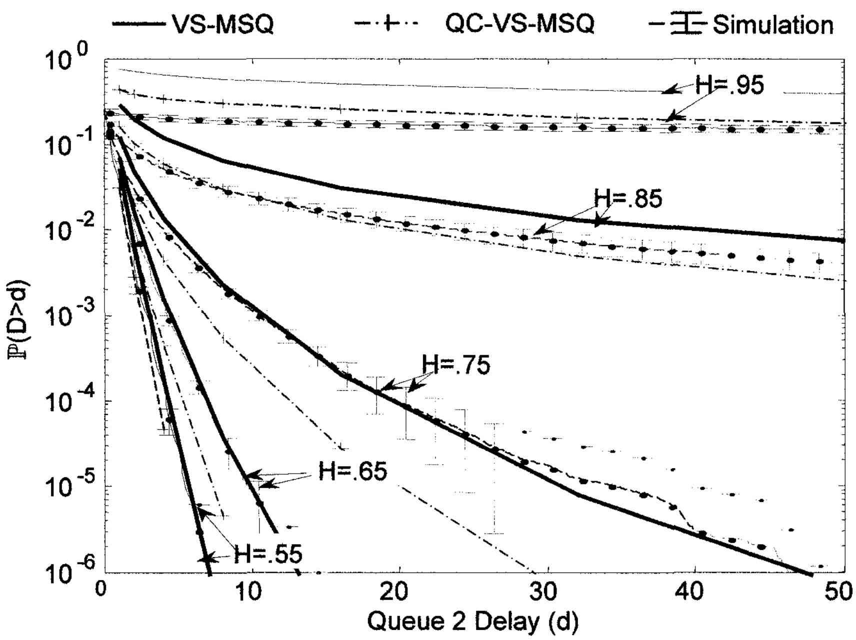


Figure 4-18: Queuing delay survivor function at a load of 70% for different service rate self-similarity for queue 2

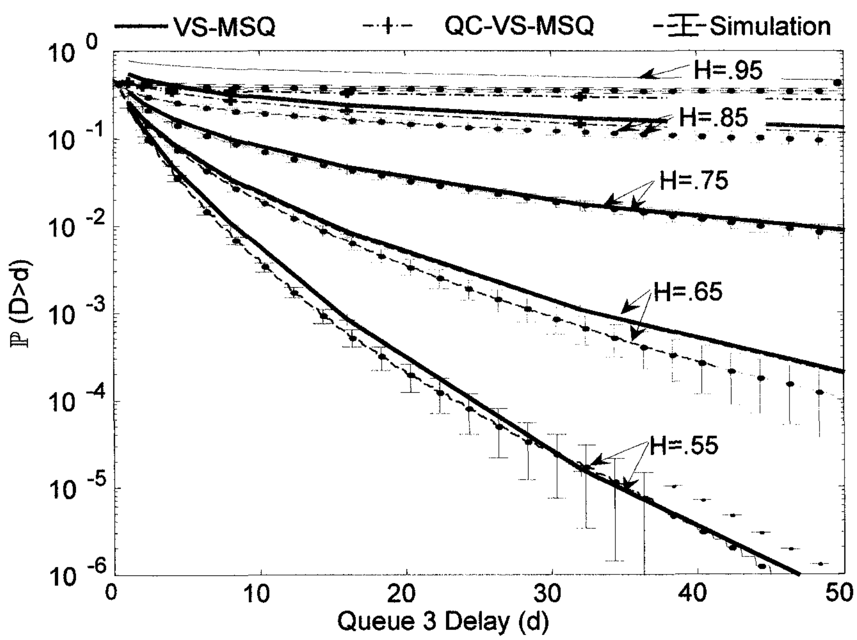


Figure 4-19: Queuing delay survivor function at a load of 70% for different service rate self-similarity for queue 3

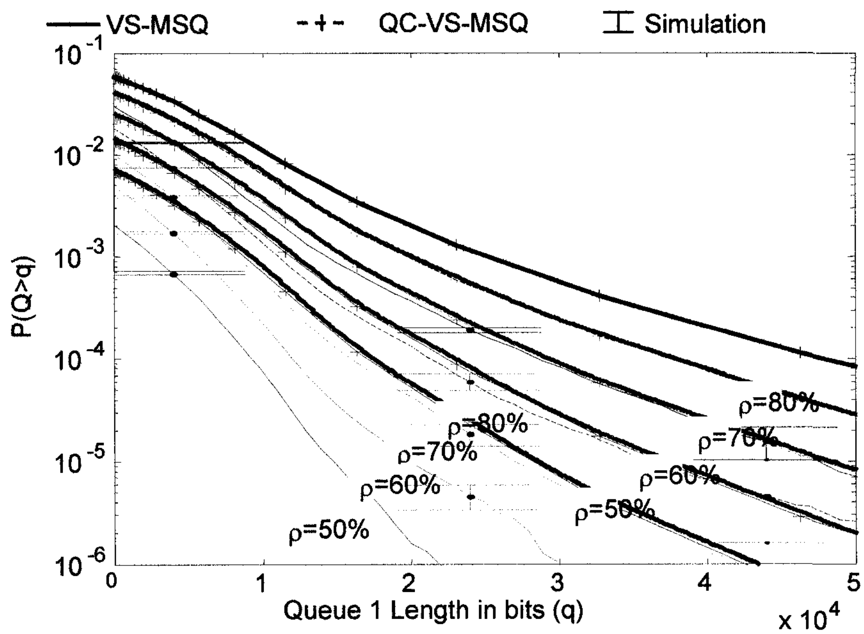


Figure 4-20: Queue length survivor function at a load of 70% for different service rate self-similarity for queue 1

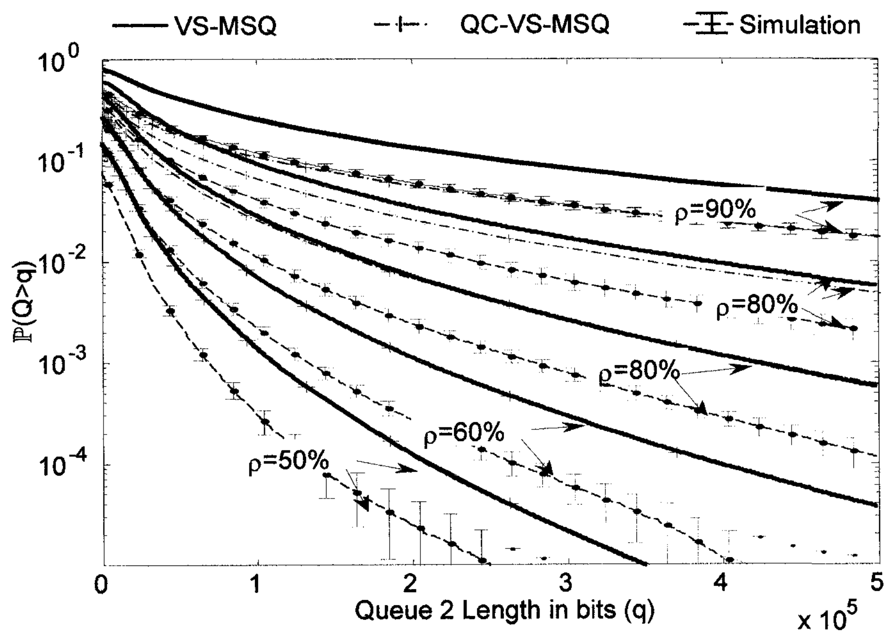


Figure 4-21: Queue length survivor function at a load of 70% for different service rate self-similarity for queue 2

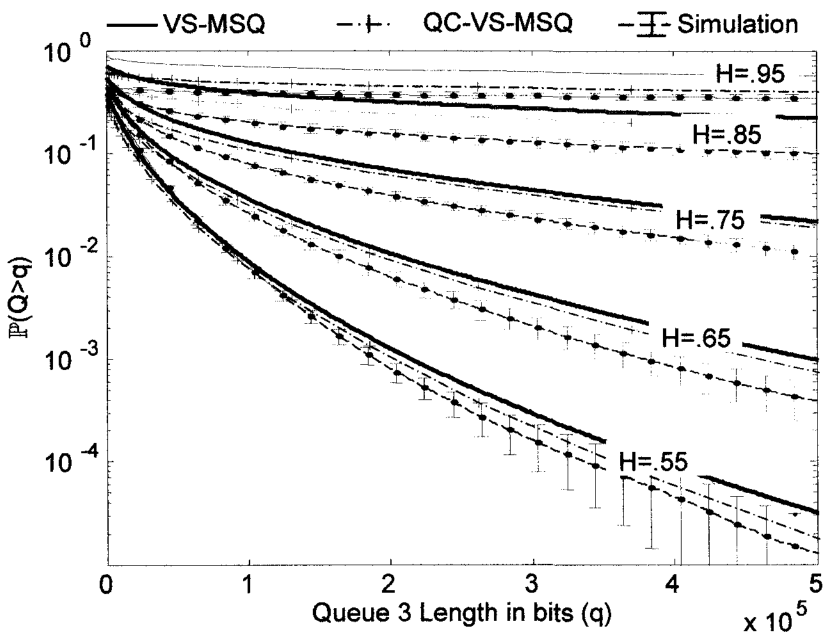


Figure 4-22: Queue length survivor function at a load of 70% for different service rate self-similarity for queue 3

4.4. Multi-Scale Weighted Fair Queue Analysis

In a GPS system with V queues and a total capacity of $C\,[\tau]$ during a time interval τ, the minimum link capacity allocated to queue v is proportional to its assigned weight, w_v, i.e.,

$$C_v[\tau]=\frac{w_v C[\tau]}{\sum_{i=1}^{V} w_i}. \tag{4.26}$$

The above capacity allocation is independent of the queue status and no queue decomposition is needed for its calculation. Hence, we refer to this capacity as zero decomposition level capacity. The actual service rate $\hat{C}_v[\tau]$ seen by the queue v when the traffic input is equal to $K_v[\tau]$ is represented by

$$\widehat{C}_v[\tau]=\frac{w_v C[\tau]}{\sum_{i=1}^{V}\phi_i w_i}, \tag{4.27}$$

where $\phi = \{\phi_j \mid j = 1,,2,\dots,V\}$ is set of the queue states, i.e., $\phi_j = 1$ if $Q_j > 0$ and $\phi_j = 0$ if $Q_j = 0$. The service rate $\hat{C}_v[\tau]$ can have a range of values defined by

$$C_v[\tau] \le \hat{C}_v[\tau] \le C\ [\tau] \tag{4.28}$$

The actual value of $\hat{C}_v[\tau]$ will depend on the remaining service rates from other queues. A lower bound on the remaining service $R_v[\tau]$ from queue v can be obtained as:

$$R_v[\tau] \ge \max\left(0, C_v[\tau] - K_v[\tau]\right) \tag{4.29}$$

It is noted that this lower bound is only tight at the smallest time-scale. In a GPS system with V queues, the conditions in (4.29) introduce 2^{V-1} queue states, Ω_i, seen by each queue. For any queue v, the service seen by queue v will depend on the status of the other queues Ω. In order to explain this coupling, we consider a simple example of three queues at the beginning of a given time scale τ, where the values of $K_1[\tau], K_2[\tau]$ and $K_3[\tau]$ are 4,6 and 12 (units), and the queue assigned weights are w_1=0.5, w_2=0.3 w_3=0.2 respectively. Assuming that these queues are served using an ideal bit-by-bit GPS scheme at a rate of $C[\tau]/\tau = 20/\tau$. The queue length and service received by each queue are shown in Figure 4-23. During the interval τ the queue goes through three stages. During the time form 0 to 0.4 τ each queue gets a service rate proportional to its weight as defined by (4.26) because all the queues are active. During the second stage that starts after 0.4τ, capacity that is not used by the inactive queue 1 will be shared between queues 2 and 3 according to (4.27). When queue 2 becomes inactive at 0.7 τ, only queue 3 remains active so it will get the whole service rate of 20/τ, which will cause its length at the end of τ to be equal to 2. The service rate decoupling in this example went through three stages. The available service $\hat{C}_v[\tau]$ to each queue v during each stage is shown in Figure 4-23. For an active queue, there are many possibilities for the stages that the system can go through during a time interval τ. However, since this thesis deals with the multi-scale characterization of the traffic, it is only interested in the total service available to each queue during the interval τ rather than the actual states that the queue goes through to receive this service. Using Figure 4-23 it can be seen that $\hat{C}_1[\tau], \hat{C}_2[\tau]$ and $\hat{C}_3[\tau]$ in this example are equal to 4, 6, 10 respectively.

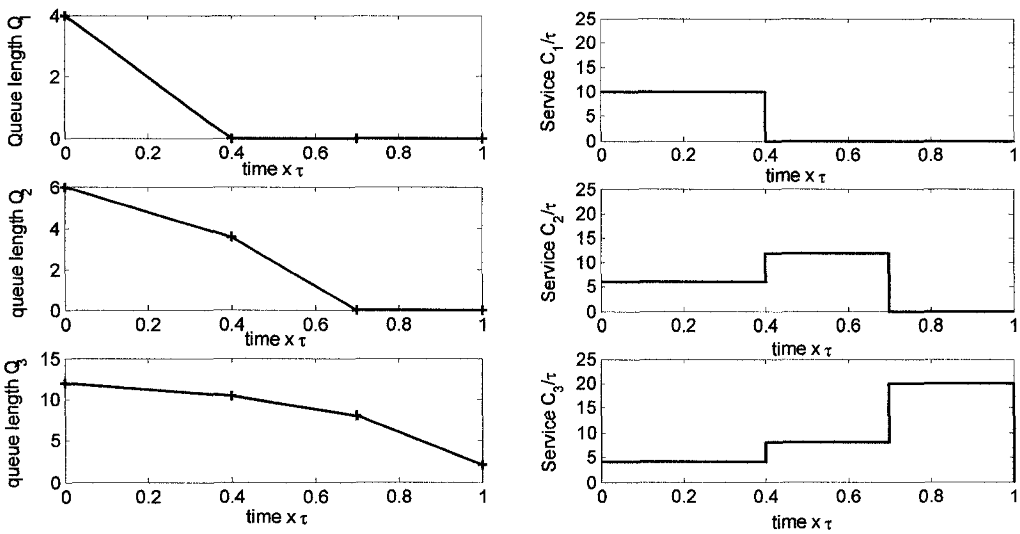


Figure 4-23 : An example service sequence during a time interval τ

In order to explain how the proposed approach evaluates the decoupled service rate seen by each queue, we first consider the simplest case of a two-queue GPS system. Considering queue 1, the decoupled service $\hat{C}_1[\tau]$ is larger than $C_1[\tau]$ if and only if $K_2[\tau]<C_2[\tau]$. When $K_2[\tau]<C_2[\tau]$ the remaining service $R_2[\tau]=\max(C_2[\tau]-K_2[\tau],0)$ is added to $C_1[\tau]$ and, the decoupled service rate queue 1 can then be obtained as follows:

$$\hat{C}_1[\tau]=C_1[\tau]+R_2[\tau]. \tag{4.30}$$

Similarly, for queue 2, we have

$$\hat{C}_2[\tau]=C_2[\tau]+R_1[\tau]. \tag{4.31}$$

In the 2-queue example, only one level of queue decoupling is needed because the remaining service rate from $\hat{C}_1[\tau]$ and $\hat{C}_2[\tau]$ can not be used by queues 2 and 1 respectively. The queues can not make use of this remaining capacity because it exists only when both queues are inactive and hence they do not have any data to transmit. .

Unlike the 2-queue case, an additional level of queue decoupling will appear in the 3-queue case. The two levels of decoupling for $v=1$ and the states of the remaining service Ω and the state of the queue activity ϕ^+ at the end of the time interval τ are shown in Figure 4-24. When the queue is in state Ω_1, no service remains from both queues 2 and 3, and hence queue 1 is left with a zero-level service of $C_1[\tau]$ as calculated by (4.26). In state Ω_2, the remaining service rate from $R_2[\tau]+R_3[\tau]$ is added to the zero-level decoupled service $C_1[\tau]$. Since both queues 2 and 3 become inactive, they do not present any further interaction with queue 1 and hence no further decoupling is required.

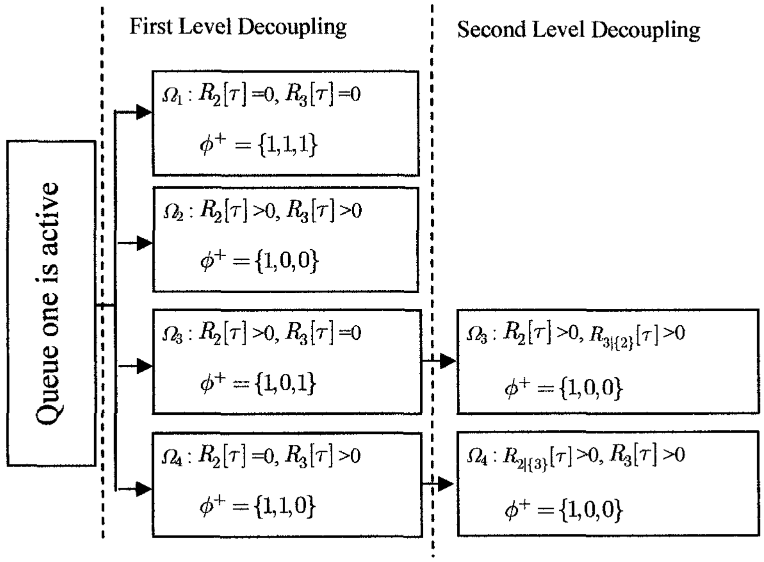


Figure 4-24: Decomposition levels for a 3-queue GPS system

In state Ω_3, queue 2 uses part of its service rate and becomes inactive leaving only queues 1, 3 active. Because queue 3 remains active, queues 1 and 3 can be treated as a 2-queue GPS system with the inputs, $K_1[\tau]$ and $(K_3[\tau] | K_3[\tau] > C_3[\tau])$, and the overall capacity of $(1-w_2)C\ [\tau]+R_2[\tau]$. As a result, queue 1 obtains a share of this capacity equal to $C_1[\tau]+w_1R_2[\tau]/(w_1+w_3)$, while queue 3 obtains a share proportional to $C_3[\tau]+w_3R_2[\tau]/(w_1+w_3)$. The value of $C_3[\tau]+ w_3R_2[\tau]/(w_1+w_3)$ may exceed $K_3[\tau]$, causing a new level of remaining service from queue 3 to appear. This second-level remaining service can only be used by queue 1 because in this case both queue 2 and 3 have become inactive. Similar arguments are applied to state Ω_4 for queues 1 and 2. Let $R_{v1|\{v2,v3....\}}$ denote the remaining service rate from queue $v1$ when its services rate is increased by the remaining service rates from queues $v2, v3, \ldots$. Then the second-level remaining bandwidth that can be used by queue 1 is

$$R_{2|\{3\}}[\tau] = \max\left(0, C_2[\tau] + \frac{w_2 R_3[\tau]}{(w_1 + w_2)} - K_2[\tau] \middle| K_2[\tau] > C_2[\tau]\right) \tag{4.32}$$

$$R_{3|\{2\}}[\tau] = \max\left(0, C_3[\tau] + \frac{w_3 R_2[\tau]}{(w_1 + w_3)} - K_3[\tau] \middle| K_3[\tau] > C_3[\tau]\right) \tag{4.33}$$

The total distribution function of the service rate $\hat{C}_1[\tau]$ for queue 1 when $R_1[\tau] = 0$, is characterized by

$$\|(\hat{C}_1[\tau] < c) = \sum_i \|(\Omega_i)\|(\Gamma_i + w_1 C[\tau] < c), \tag{4.34}$$

where,

$$\Omega_1 : R_2[\tau] = 0, R_3[\tau] = 0,\ \Gamma_1 = 0 \tag{4.35}$$

$$\Omega_2 : R_2[\tau] > 0, R_3[\tau] > 0,\ \Gamma_2 = R_3[\tau] + R_2[\tau] \tag{4.36}$$

$$\Omega_3 : R_2[\tau] > 0, R_3[\tau] = 0,\ \Gamma_3 = R_{3|\{2\}}[\tau] + w_1 R_2[\tau] / (w_1 + w_3) \tag{4.37}$$

$$\Omega_4 : R_2[\tau] = 0, R_3[\tau] > 0,\ \Gamma_4 = R_{2|\{3\}}[\tau] + w_1 R_3[\tau] / (w_1 + w_2). \tag{4.38}$$

The evaluation of the 3-queue system, shows that for a system of V queues, each queue will need $(V-1)$ levels of remaining capacity calculation in order to determine all the remaining capacity it can use from the other queues. Similarly, in the four-queue system, the capacity distribution available to queue 1 is

$$\Omega_1 : R_2[\tau] = 0, R_3[\tau] = 0, R_4[\tau] = 0, \Gamma_1 = 0 \tag{4.39}$$

$$\Omega_2 : R_2[\tau] > 0, R_3[\tau] > 0, R_4[\tau] > 0, \Gamma_2 = R_2[\tau] + R_3[\tau] + R_4[\tau] \tag{4.40}$$

$$\begin{aligned}&\Omega_3 : R_2[\tau] > 0, R_3[\tau] > 0, R_4[\tau] = 0,\\&\Gamma_3 = R_{4|\{2,3\}}[\tau] + w_1(R_2[\tau] + R_3[\tau]) / (w_1 + w_4)\end{aligned}, \tag{4.41}$$

$$\begin{aligned}&\Omega_4 : R_2[\tau] > 0, R_3[\tau] = 0, R_4[\tau] > 0,\\&\Gamma_4 = R_{3|\{2,4\}}[\tau] + w_1(R_2[\tau] + R_4[\tau]) / (w_1 + w_3)\end{aligned}, \tag{4.42}$$

$$\begin{aligned}&\Omega_5 : R_2[\tau] = 0, R_3[\tau] > 0, R_4[\tau] > 0,\\&\Gamma_5 = R_{2\{3,4\}}[\tau] + w_1(R_3[\tau] + R_4[\tau]) / (w_1 + w_2)\end{aligned}, \tag{4.43}$$

$$\begin{aligned}&\Omega_6 : R_2[\tau] > 0, R_3[\tau] = 0, R_4[\tau] = 0,\\&\Gamma_6 = w_1 R_{3|\{2\}}[\tau] / (w_1 + w_4) + w_1 R_{4|\{2\}}[\tau] / (w_1 + w_3)\\&\quad + R_{3|\{4|\{2\}\}}[\tau] + R_{4|\{3|\{2\}\}}[\tau] + w_1 R_2[\tau] / (w_1 + w_3 + w_4)\end{aligned} \tag{4.44}$$

$$\begin{aligned}&\Omega_7 : R_2[\tau] = 0, R_3[\tau] > 0, R_4[\tau] = 0,\\&\Gamma_7 = w_1 R_{2|\{3\}}[\tau] / (w_1 + w_4) + w_1 R_{4|\{3\}}[\tau] / (w_1 + w_2)\\&\quad + R_{2|\{4|\{3\}\}}[\tau] + R_{4|\{2|\{3\}\}}[\tau] + w_1 R_3[\tau] / (w_1 + w_2 + w_4)\end{aligned} \tag{4.45}$$

$$\begin{aligned}&\Omega_8 : R_2[\tau] = 0, R_3[\tau] = 0, R_4[\tau] > 0,\\&\Gamma_8 = w_1 R_{2|\{4\}}[\tau] / (w_1 + w_3) + w_1 R_{3|\{4\}}[\tau] / (w_1 + w_2)\\&\quad + R_{2|\{3|\{4\}\}}[\tau] + R_{3\{2|\{4\}\}}[\tau] + w_1 R_4[\tau] / (w_1 + w_2 + w_3)\end{aligned} \tag{4.46}$$

The same approach can be used to obtain the service rate for each queue given the remaining bandwidth of the other queues. The second-level decomposition involves terms like $R_{2|\{3\}}[\tau]$, $R_{3|\{2\}}[\tau]$ and $R_{4|\{2,3\}}[\tau]$ in (4.37), (4.38) and (4.41), respectively. The third-level decomposition involves terms like $R_{4|\{3|\{2\}\}}$ and $R_{2|\{4|\{3\}\}}$ (4.45) where $R_{4|\{3|\{2\}\}}$ represents the remaining bandwidth from queue 4 when its service rate is increased by the remaining service rate from queue 3, given that queue 3 service rate is increased by the remaining service rate from queue 2. Equation (4.34) reflects the complicated interdependencies between queues in GPS system. Most queue decoupling approaches only consider the firsts two terms corresponding to Ω_1 and Ω_2. More accurate techniques consider all the queue states involved in the first-level decomposition, i.e., including only terms like $R_2[\tau]$ and $R_3[\tau]$. Considering only the first-level remaining service rate means that $\mathbb{P}(\widehat{C}_1[\tau] < c)$ would be pessimistic (i.e., high probability will be given to lower service rate). This will cause an overestimation of the queue length and queue delay survivor functions. As shown in (4.34), the significance of each term depends on its probability of Ω_2, e.g., in the 3-queue case, when $\mathbb{P}(R_2[\tau]=0)= \mathbb{P}(R_3[\tau]=0) = 0.5$ all terms in (4.34), will have significant contribution to the queue survivor function, while if $\mathbb{P}(R_2[\tau]=0)= \mathbb{P}(R_3[\tau]=0)=0.99$ then the effect of the first term becomes more dominant and the other terms will cause minor errors to the calculation. The impact of this effect will depend on the value of $\mathbb{P}(\Gamma_i + w_1 C[\tau])$.

For a *constant* service rate, the value of $C[\tau]$ is fixed over each time scale. In this case, the pdf of $R_v[\tau]$ can be evaluated as

$$f_{R_v[\tau]}(r) = f_{K_v[\tau]}(C_v[\tau] - r), 0 \le r \le C_v[\tau] \tag{4.47}$$

where, $f_{K_v[\tau]}$ is the *pdf* of the arrival $k_v[\tau]$. Because the traffic arrivals are independent of each other, the remaining service rates are also independent of each other, which simplify the evaluation of (4.34) in the constant service case.

Evaluation of terms in (4.34) becomes more complicated when $C[\tau]$ is variable. For example, the values of $R_3[\tau]$ and $R_2[\tau]$ in (4.36) change with $C[\tau]$. Since the remaining services from these queues depend on the offered capacity $C[\tau]$ and its distribution, they depend on each other and also on the service rate. For example, the remaining service of queue v at the instant when the service rate is equal to c, has to be smaller than $w_v c$. When added to the service rate of another queue u, it can only be added to exactly the value $w_u c$ and cannot be added to any other value of the random process $w_v C[\tau]$. Taking the term corresponding to (4.36) in the 3-queue expansion of (4.34) as an example, we have

$$\|(\Omega_2)\|(\Gamma_2+w_1C[\tau]<c)=\int_{x=0}^{c/w_1}(\|(\Omega_2)\|(\Gamma_2+w_1C[\tau]<c)|C[\tau]=x)f_{C[\tau]}(x)dx \tag{4.48}$$

where $f_{C[\tau]}$ is the *pdf* of service rate $C[\tau]$ and $(\Gamma_2+w_1C[\tau]<c)|C[\tau]=x)$ refers to the evaluation of the remaining service rate when the service rate $C[\tau]$ is equal to x.

To simplify the calculation and to provide an understanding of the operation of the queue, we approximate the distribution describing terms in (4.34) by fitting them to a Beta distribution using mean-variance fitting. As an illustrative example, we consider the fitting of these distributions for queue 1 in a two-queue system as described by (4.30) and (4.31). The service rate seen by queue 1 can be described by two distributions, one for $C_1[\tau]$ when $R_2[\tau]=0$ and the other for $C_1[\tau]+R_2[\tau]$ when $R_2[\tau]>0$. The first and second moments are then evaluated as

$$\mathbb{E}\left(\hat{C}_1[\tau]\|C[\tau]=x,R_2[\tau]>0\right)=x-\mathbb{E}\left(K_2[\tau]|K_2[\tau]<w_2x\right), \tag{4.49}$$

$$\begin{aligned}\mathbb{E}\left(\hat{C}_1[\tau]^2|C[\tau]=x,R_2[\tau]>0\right)&=\mathbb{E}\left(\left(x-K_2[\tau]\right)^2|K_2[\tau]<w_2x\right)\\&=x^2+2\mathbb{E}(K_2[\tau]|K_2[\tau]<w_2x)x+\mathbb{E}(K_2^2[\tau]|K_2[\tau]<w_2x)\end{aligned} \quad ,(4.50)$$

$$\begin{aligned}\text{Where, } \mathbb{E}\left(K_2[\tau]|K_2[\tau]<w_2x\right)&=\Lambda_2^{(K)}\int_0^{x_2^*}y\left(\frac{y^{a_{K_2[\tau]}-1}(1-y)^{b_{K_2[\tau]}-1}}{\beta\left(a_{K_2[\tau]},b_{K_2[\tau]},x_2^*\right)}\right)dy \quad ,\\&=\Lambda_2^{(K)}\frac{a_{K_2[\tau]}}{a_{K_2[\tau]}+b_{K_2[\tau]}}\frac{\beta\left(a_{K_2[\tau]}+1,b_{K_2[\tau]},x_2^*\right)}{\beta\left(a_{K_2[\tau]},b_{K_2[\tau]},x_2^*\right)}\end{aligned}$$

and $x_2^*=w_2x/\Lambda_2^{(K)}$. Similarity,

$$\mathbb{E}\left(K_2^2[\tau]|K_2[\tau]<w_2x\right)=\frac{\left(\Lambda_2^{(K)}\right)^2b_{K_2[\tau]}(b_{K_2[\tau]}+1)}{(a_{K_2[\tau]}+b_{K_2[\tau]})(a_{K_2[\tau]}+b_{K_2[\tau]}+1)}\frac{\beta\left(a_{K_2[\tau]}+2,b_{K_2[\tau]},x_2^*\right)}{\beta\left(a_{K_2[\tau]},b_{K_2[\tau]},x_2^*\right)}.$$

where a_{k2}, b_{k2} are the beta parameter characterizing $K_2[\tau]$. It follows that

$$\mathbb{E}\left(\hat{C}_1[\tau] \middle| R_2[\tau]>0\right)=\frac{\Lambda_2^{(K)}}{\beta(a_{K_2[\tau]},b_{K_2[\tau]},C_2^*)}\int_{x=0}^{C}\left(\frac{x}{\Lambda_2^{(K)}}-\frac{a_{K_2[\tau]}}{a_{K_2[\tau]}+b_{K_2[\tau]}}\frac{\beta\left(a_{K_2[\tau]}+1,b_{K_2[\tau]},x_2^*\right)}{\beta\left(a_{K_2[\tau]},b_{K_2[\tau]},x_2^*\right)}\right)f_C(x)dx, \quad (4.51)$$

where, $C_2^* = w_2 C[\tau] / \Lambda_2^{(K)}$,

$$\mathbb{E}\left(\hat{C}_1[\tau]^2 \middle| R_2[\tau]>0\right)=$$

$$\frac{\left(\Lambda_2^{(K)}\right)^2}{\beta(a_{K_2[\tau]},b_{K_2[\tau]},C_2^*)}\int_{x=0}^{C}\left(\begin{array}{c}\frac{x^2}{\left(\Lambda_2^{(K)}\right)^2}-\frac{2x}{\Lambda_2^{(K)}}\frac{a_{K_2[\tau]}}{a_{K_2[\tau]}+b_{K_2[\tau]}}\frac{\beta\left(a_{K_2[\tau]}+1,b_{K_2[\tau]},x_2^*\right)}{\beta\left(a_{K_2[\tau]},b_{K_2[\tau]},x_2^*\right)})+\\ \frac{b_{K_2[\tau]}(b_{K_2[\tau]}+1)}{(a_{K_2[\tau]}+b_{K_2[\tau]})(a_{K_2[\tau]}+b_{K_2[\tau]}+1)}\frac{\beta\left(a_{K_2[\tau]}+2,b_{K_2[\tau]},x_2^*\right)}{\beta\left(a_{K_2[\tau]},b_{K_2[\tau]},x_2^*\right)}\end{array}\right)f_C(x)dx \quad (4.52)$$

$$\mathbb{E}\left(\hat{C}_1[\tau] \middle| R_2[\tau]=0\right)=\int_{x=0}^{C}\frac{x\left(1-\beta(a_{K_2[\tau]},b_{K_2[\tau]},x_2^*\right)}{1-\beta\left(a_{K_2[\tau]},b_{K_2[\tau]},C_2^*\right)}f_C(x)dx, \text{ and}$$

$$\mathbb{E}\left(\hat{C}_1[\tau]^2 \middle| R_2[\tau]=0\right)=\int_{x=0}^{C}\frac{(x)^2\left(1-\beta(a_{K_2[\tau]},b_{K_2[\tau]},x_2^*\right)}{1-\beta\left(a_{K_2[\tau]},b_{K_2[\tau]},C_2^*\right)}f_C(x)dx$$

The parameters $a_{\hat{C}_1}$ and $b_{\hat{C}_1}$ characterizing the Beta *pdf* can be evaluated using

$$a_{\hat{C}_1[\tau]}=\frac{\mathbb{E}(\hat{C}_1[\tau])}{\Lambda^{(C)}}\frac{\left(\mathbb{E}(\hat{C}_1[\tau])-\mathbb{E}(\hat{C}_1^2[\tau])\right)}{\mathbb{E}(\hat{C}_1^2[\tau])-\mathbb{E}(\hat{C}_1[\tau])^2}, \text{ and} \quad (4.53)$$

$$b_{\hat{C}_1[\tau]}=a_{\hat{C}_1}\left(\mathbb{E}(\hat{C}_1[\tau])/\Lambda^{(C)}-1\right)\left(\mathbb{E}(\hat{C}_1[\tau])/\Lambda^{(C)}\right)^{-1}. \quad (4.54)$$

The same approach is used to calculate the distribution of the service rate when the second-level remaining bandwidth such as $R_{3|\{2\}}[\tau]$ is involved. In this case,

$$\begin{aligned}\mathbb{E}\left(\hat{C}_1[\tau]\middle|C=x,R_2[\tau]>0\right)&=\frac{w_1}{(w_1+w_3)}\left(x-\mathbb{E}\left(k_2[\tau]\,|\,k_2[\tau]<w_2x\right)\right)\\&+\mathbb{E}\left(\frac{w_3\hat{R}_2}{(w_1+w_3)}-k_3[\tau]\middle|w_3x[\tau]<k_3[\tau]<\frac{w_3\hat{R}_2}{(w_1+w_3)}\right)\end{aligned}\tag{4.55}$$

$$\begin{aligned}\mathbb{E}\left(\hat{C}_1^2[\tau]\middle|C=x,R_2[\tau]>0\right)&=\mathbb{E}\left(\hat{C}_1[\tau]\middle|C=x,R_2[\tau]>0\right)^2\\&+\left(\frac{w_1}{w_1+w_3}\right)^2\left(\mathbb{E}\left(K_2^2[\tau]\,|\,(k_2[\tau]<w_2x)\right)-\mathbb{E}\left(K_2[\tau]\,|\,(K_2[\tau]<w_2x)\right)^2\right)\\&+\mathbb{E}\left(\left(\frac{w_3}{w_1+w_3}\hat{R}_2-K_3[\tau]\right)^2\,\middle|\,w_3x<K_3[\tau]<\frac{w_3}{w_1+w_3}\hat{R}_2\right)\\&-\mathbb{E}\left(\left(\frac{w_3}{w_1+w_3}\hat{R}_2-K_3[\tau]\right)\middle|\,w_3x<K_3[\tau]<\frac{w_3}{w_1+w_3}\hat{R}_2\right)^2\end{aligned}\tag{4.56}$$

where, $\hat{R}_2=x-(K_2[\tau]\,|\,K_2[\tau]<w_2x)$.

Using the Beta-distribution approximation, two distributions will be needed to describe the service rate Γ_3 in the three-queue case. It would require one distribution to describe the service rate for $R_{3|\{2\}}[\tau]=0$ and another for $R_{3|\{2\}}[\tau]>0$. This will split Ω_3 in (4.37) into $\Omega_{3(1)}$ and $\Omega_{3(2)}$ where,

$$\begin{aligned}&\Omega_{3(1)}:R_2[\tau]>0,R_3[\tau]=0,k_3[\tau]>\hat{R}_2w_3/(w_1+w_3),\\&\Gamma_{3(1)}=w_1R_2[\tau]/(w_1+w_3)\end{aligned}\tag{4.57}$$

$$\begin{aligned}&\Omega_{3(2)}:R_2[\tau]>0,R_3[\tau]=0,w_3x<K_3[\tau]<\hat{R}_2w_3/(w_1+w_3),\\&\Gamma_{3(2)}=R_{3|\{2\}}[\tau]+w_1R_2[\tau]/(w_1+w_3)\end{aligned}\tag{4.58}$$

The condition $w_3 x < k_3[\tau] < (w_3 / (w_1 + w_3))\hat{R}_2$ complicates the evaluation of $\mathbb{E}\left(w_3 / (w_1 + w_3))\hat{R}_2 - k_3[\tau] \mid \Omega_{3(2)}\right)$ in (4.55) and (4.56). To overcome this problem, we find the following Beta-distributed S by using mean-variance fitting

$$S = \left(\left(\Lambda_3^{(K)} - w_3 x\right) + w_3 \hat{R}_2 / (w_1 + w_3) - K_3[\tau] \mid w_3 x < K_3[\tau]\right),$$

where $\Lambda_3^{(K)}$ is the input traffic value at the coarsest time-scale for queue 3. The term $\Lambda_3^{(K)} - w_3 x$ ensures that S is always positive. Its first and second moments can be derived as:

$$\_\_(S) = \begin{pmatrix} \Lambda_3^{(K)} - w_3 x[\tau] - \_\_(k_3[\tau] \mid K_3[\tau] > w_3 x) \\ +(w_3 / (w_1 + w_3))\left(x - \_\_(K_2[\tau] \mid K_2[\tau] < w_2 x)\right) \end{pmatrix} \tag{4.59}$$

$$\mathbb{E}(S^2) = \begin{pmatrix} \mathbb{E}(S)^2 + \left(\mathbb{E}(K_2^2[\tau]) - \left[\mathbb{E}\left(K_2[\tau]\right)\right]^2 \mid K_2[\tau] < w_2 x\right) \\ + \left(\mathbb{E}(K_3^2[\tau]) - \left[\mathbb{E}\left(k_3[\tau]\right)\right]^2 \mid K_3[\tau] > w_3 x\right) \end{pmatrix}. \tag{4.60}$$

Subsequently, the Beta-distribution parameters, a_s and b_s, can be calculated from $\mathbb{E}(S)$ and $\mathbb{E}(S^2)$, and the terms conditioned on $w_3 x[\tau] < k_3[\tau] < \hat{R}_2$ in (4.55) and (4.56) can be evaluated as

$$\mathbb{E}\left(\hat{R}_2 w_3 / (w_1 + w_3) - K_3[\tau] \mid \Omega_{3(2)}\right) = \mathbb{E}\left(S \mid S > \left(\Lambda_3^{(K)} - w_3 x\right)\right) - \left(\Lambda_3^{(K)} - w_3 x\right) \tag{4.61}$$

$$\begin{aligned} &\mathbb{E}\left(\left(\hat{R}_2 w_3 / w_1 + w_3 - k_3[\tau]\right)^2 \middle| \Omega_{3(2)}\right) = \\ &\left(\Lambda_3^{(K)} - w_3 x\right)^2 + \left(\mathbb{E}\left(S^2\right) - 2\left(\Lambda_3^{(K)} - w_3 x\right)\mathbb{E}\left(S\right) \mid S > \Lambda_3^{(K)} - w_3 x\right) \end{aligned} \tag{4.62}$$

The same approach is applied to Γ_4, which results in a total of 6 Beta-distributed random variables to describe the 3-queue case. This means that, in some cases, the number of distributions exceeds the number of conditions of Ω. It follows that the service rate seen by a queue in a GPS system with V queues will be characterized by $\sum_{i=1}^{V-1} 2^{V-i}$ distributions. The *pdf* of $\hat{C}_1[\tau]$ can be evaluates using (4.34) as the weighted sum of these *pdf*'s where each weighting coefficient is the occurrence probability of the corresponding combination of remaining service rate. Based on these service rate approximations, the queue survivor function can be calculated using (4.5).

4.5. Illustrative Results

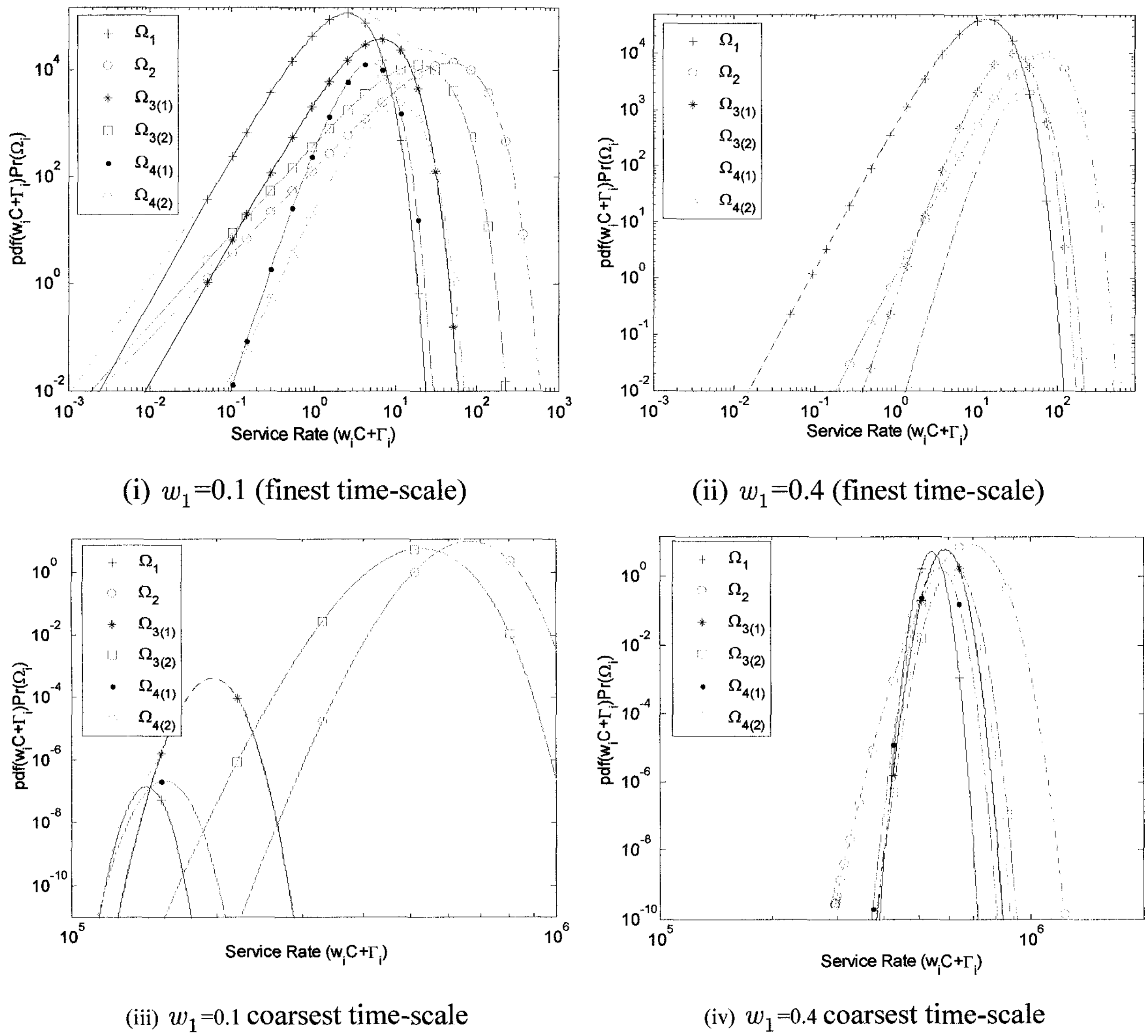
In this section, we examine the accuracy of the proposed analysis using a three-queue GPS model. Multi-scale traffic [119] is used in both the analysis and simulation so that the difference between analysis and simulation is mainly due to the inaccuracy in the analytical estimation. At the finest time scale, $\tau = 2^0 = 1$, the three traffic inputs have the same mean, μ=25, and standard deviation, σ=15. The traffic traces are modeled using a number of scales M=14, and three different Hurst parameters, H_1=0.65, H_2=0.75 and H_3=0.85 (to characterize their level of self-similarity). The various Hurst parameter values are used to investigate the capability of the analysis in capturing the coupling between the queues at different queue weights, and the effects of self-similarity of each traffic source on the *service* received by other traffic sources. The mean and standard deviation of the service rate are set to $V\mu / \rho$ and $V\sigma / \rho$, respectively, where ρ is the traffic loading and V is the number of queues. Results shown in this thesis are for a high load of ρ=0.85 to ensure that all the queues will have to share the service rate and hence provide a good test of the proposed analytical model. The Hurst parameter of the service rate is set to 0.75. The weight of the third queue w_3 is set to 0.3 while queue 2 weight w_2 is varied from 0.05 to 0.65. Therefore, the weight of the first queue is w_1=0.7-w_2. The weight of queue 3 is fixed and $w_3\mathbb{E}(C)/\rho > \mu_3$ to examine the effects of the other two queue weight assignment on the performance of fixed-weight queue.

4.5.1. Probability distribution of $\hat{C}_v$

As shown in (4.34), the probability distribution function of the service rate of queue 1, $\widehat{C_1}[\tau]$, is the sum of all $\mathbb{P}(\Omega_i) f_{(\Gamma_i + w_1 C[\tau])}(x)$. Figure 4-25 shows all $\mathbb{P}(\Omega_i) f_{(\Gamma_i + w_1 C[\tau])}(x)$ and their sum (shown in dashed line) in a 3-queue system with w_1=0.1 and w_1=0.4 at the finest and coarsest time-scales, where $f_{(\Gamma_i + w_1 C[\tau])}(x)$ is the pdf of $\Gamma_i + w_1 C[\tau]$.

At the finest time-scale, for both weights, the highest $\mathbb{P}(\Omega_i)f_{(\Gamma_i+w_1C[\tau])}(x)$ corresponds to Ω_1 in Figure 4-25 (i). For w_1=0.1, higher values of service are added by terms corresponding to Ω_2 and Ω_3 in (4.36) and (4.37). The pdf*'s* corresponding to both terms get their values from the remaining service rate of queue 2 because it has the highest weight. For w_1=0.4 in Figure 4-25 (ii) the other queues have the same weight, $w_2 = w_3$ =0.3 and hence, the pdf's corresponding Ω_3 and Ω_4 are very close to each other. Because w_2 and w_3 at the high load of ρ=0.85 provide a zero-level service rate that is slightly higher than their average traffic arrival, they can only provide very little added service rate to the other queues. Most of the added service rates in Figure 4-25 (ii) occurs at state Ω_2, when both queues 1 and 2 have remaining service rates. This may explain why reasonable performance bound can be obtained in rate-proportional approaches and queue-decomposition approaches that only consider Ω_2, especially in short range dependent traffic where most of the service variation comes from the finer time-scale. At the coarsest scale in Figure 4-25 (iii) the pdfs corresponding to Ω_2 and $\Omega_{3(2)}$ dominate the others, while zero-level service rate (i.e., Ω_1) contributes the least to the overall service pdf. Feasible ordering would have been able to consider the service added by Ω_2, while the first-level decoupling in [112] would have considered $\Omega_{3(1)}$ and $\Omega_{4(1)}$. However, in both cases, $\Omega_{3(2)}$ is not considered. This would make any estimation inaccurate, and hence, show the need for second-level decoupling. It is also interesting to notice that the importance of $\Omega_{3(2)}$ is not obvious at the fine scale where the feasible ordering mostly operates, showing the advantage provided by the multi-scale temporal decoupling. In Figure 4-25 (iv), all components have very close contribution to the service rate. It is noted that the difference in the input traffic Hurst parameters of queues 2 and 3 causes the pdf's corresponding to Ω_3 and Ω_4 to be different at the coarsest time-scale. A close investigation of Figure 4-25 (iv) reveals that magnitudes of the pdf's are generally smaller for w_1=0.4 because, at this weight, no component dominates the service rate, and hence the probability of all components is reduced. The contribution level of each state Ω_i can be represented by the probability of being in this state, $\mathbb{P}(\Omega_i)$ as shown in Figure 4-26 versus the time scale index, represented by $\log(\tau)/\log(2)$, for two different values of w_1. For $w_1 = 0.1$, Figure 4-26 (i) indicates that the contribution level of states Ω_1 and $\Omega_{3(1)}$ is comparable to that of Ω_2 and $\Omega_{3(2)}$ at a lower (finer) time scale, but diminishes at higher scales. However, for a large weight, i.e.,

$w_1 = 0.4$, Ω_1, $\Omega_{3(1)}$, Ω_2 and $\Omega_{3(2)}$ have nearly the same contribution level over all time scales in Figure 4-26 (ii). This shows the need to consider this relation over all time scales in order to



(i) w_1=0.1 (finest time-scale)

(ii) w_1=0.4 (finest time-scale)

(iii) w_1=0.1 coarsest time-scale

(iv) w_1=0.4 coarsest time-scale

Figure 4-25: $\mathbb{P}(\Omega_i)p(\Gamma_i + w_1 C[\tau])$ and their sum (shown in dashed line) at the finest and coarsest time-scale in a 3-queue system for w_1=0.1 and w_1=0.4

accurately characterize the decoupled service rate of each queue.

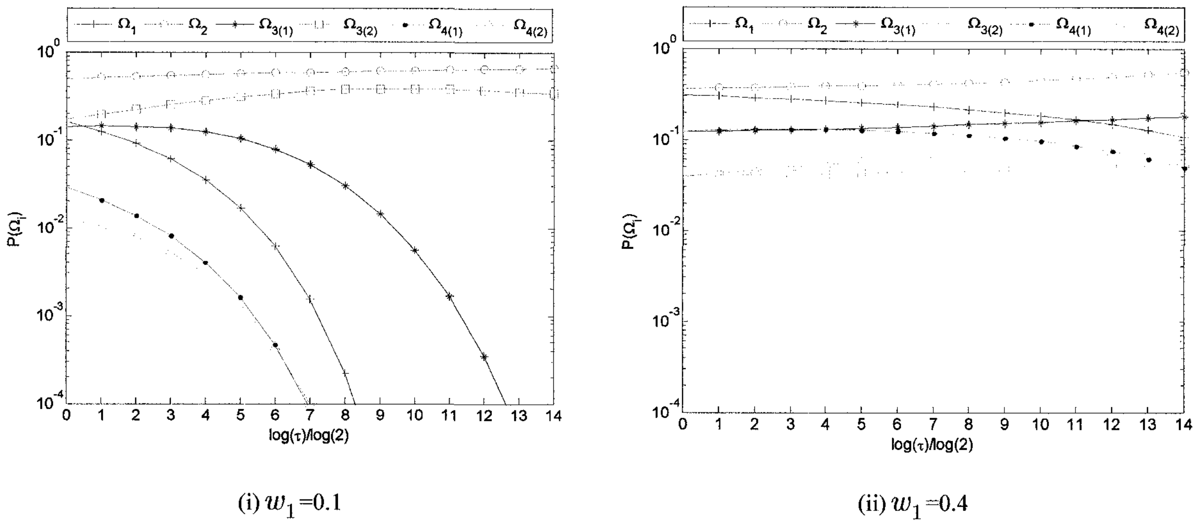


Figure 4-26: $\mathbb{P}(\Omega_i)$ **for two different weights**

4.5.2. Queue-length and Delay Survivor Functions

Figure 4-27 and Figure 4-28 compare the results on queue length and delay survivor functions obtained by simulation (dotted-line) and both analytical techniques: without (dashed-line, VS-MSQ) and with (solid-line, MC-VS-MCQ) queue-length coupling consideration. Each simulation point represents the average of 30 simulation runs with a 95% confidence interval. The results show that the proposed approach can provide an accurate estimation of the service rate, and hence, of both the queue length and delay survivor functions, i.e., the QC-VS-MCQ solid-line curves are very close to the simulation dotted-line curves. As the weight of a given queue increases, its overall service rate increases. As discussed in [53], increasing the service rate decreases the effect of the queue coupling. This decrease in the queue coupling makes the VS-MSQ estimation closer to that of the QC-VS-MSQ

4.5.3. Effect of Weight Variation:

Figure 4-29 and Figure 4-30 show $\mathbb{P}(Q > 1000)$ and $\mathbb{P}(D > 20)$ for the three queues in the system versus w_2. It is interesting to see that although the weight of queue 3 is fixed to 0.3, its queue and delay survivor functions change as the weights of the other queues change. As weight w_2 changes the remaining service rate from the other queues changes, and hence the total service presented to queue 3. It is also interesting to see that queue 3 suffers the worst performance when $w_1 = w_2 = 0.35$. In this case, the remaining service rates from both queues become smaller and, accordingly, the service share of queue 3 also decreases. As the weight of the queue 1 increase, its remaining service rate increases. In the same time the weight of queue 2 decreases hence increasing the share of queue 3 from the queue 1 remaining service. The same applies when the weight of queue 2 increases. For w_1=0.1, $\mathbb{P}(\Omega_2)$ and $\mathbb{P}(\Omega_{3(2)})$ increase with time-scale,

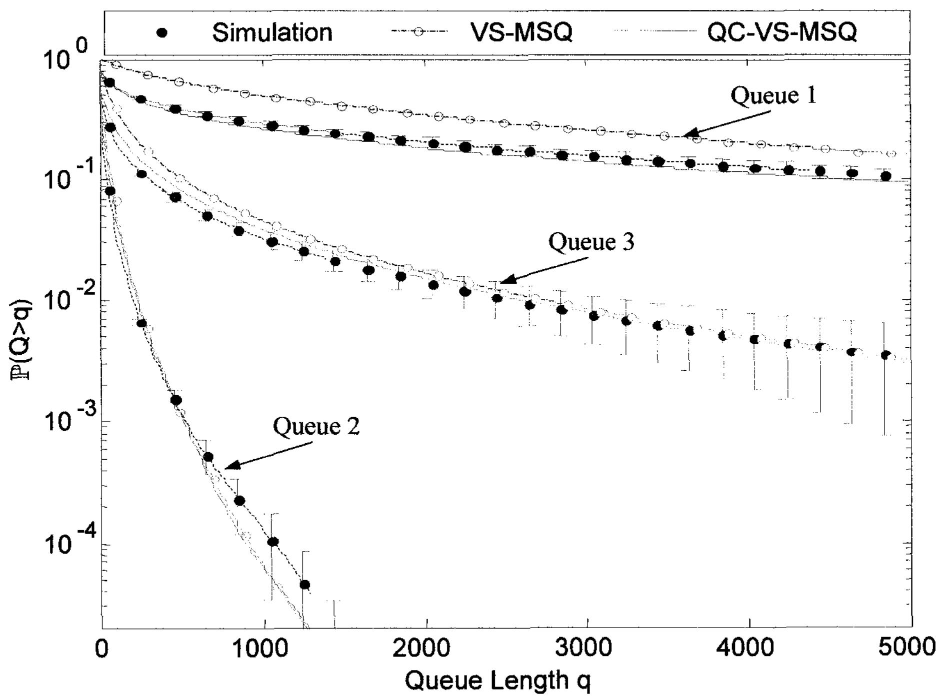


Figure 4-27: Queue-length survivor functions $w_1 = 0.1, w_2 = 0.6, w_3 = .3$

indicating the more dominant role of the remaining service at higher time scales. On the other hand, for w_1=0.4, all components maintain relatively the same ratio to each other, except for $\mathbb{P}(\Omega_1)$, $\mathbb{P}(\Omega_{3(1)})$. It is interesting that $\mathbb{P}(\Omega_{3(1)})$ and $\mathbb{P}(\Omega_{4(1)})$ are equal at lower scales because they include remaining service from queues fed with traffic traces of equal mean, variance and weight. However, because of their different self-similarity, their values change at higher time scales.

Figure 4-29 and Figure 4-30 show that the proposed technique can capture the effects of weight variation on the queue performance. Furthermore, taking the queue-length effect in consideration also improves the estimation accuracy for queues with low-weight assignment.

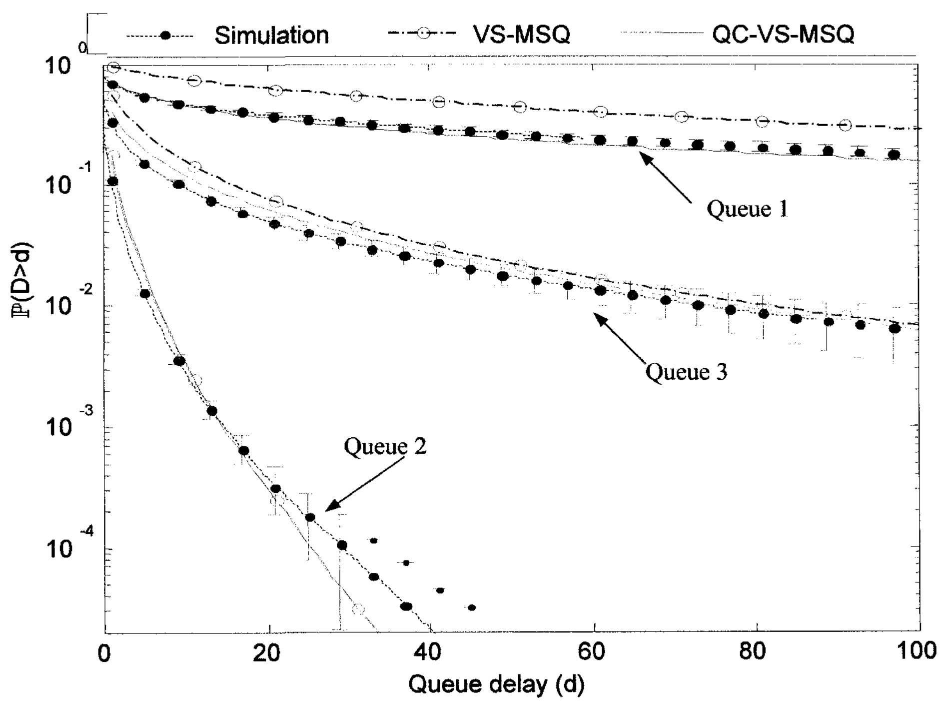


Figure 4-28: Delay survivor functions $w_1 = 0.1, w_2 = .6, w_3 = .3$

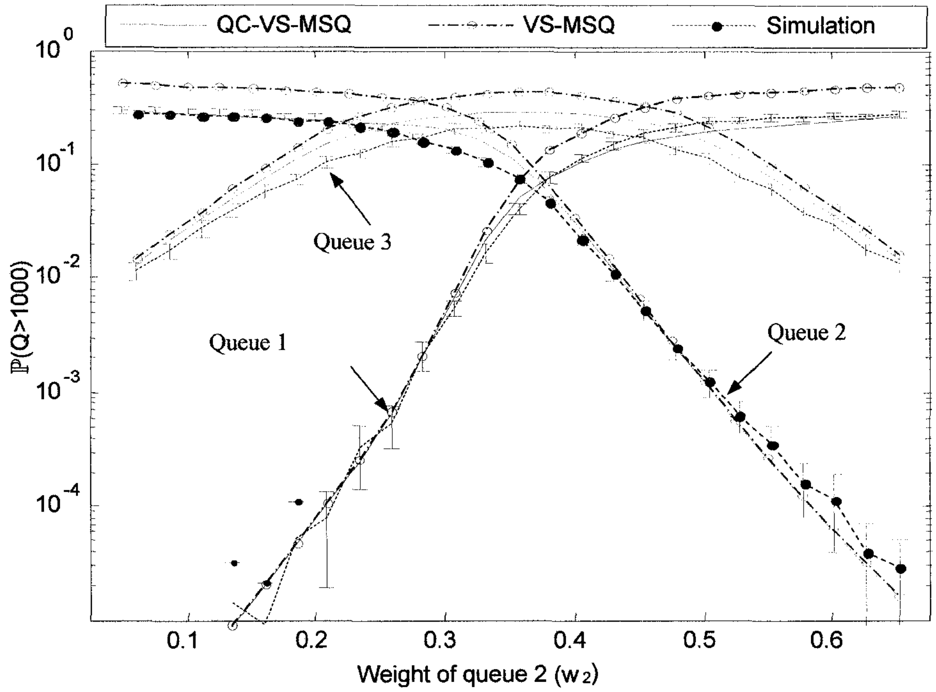


Figure 4-29: $\mathbb{P}(Q > 1000)$ **versus** w_2 **(with** $w_3 = 0.3$, $w_1 = 0.7 - w_2$**)**

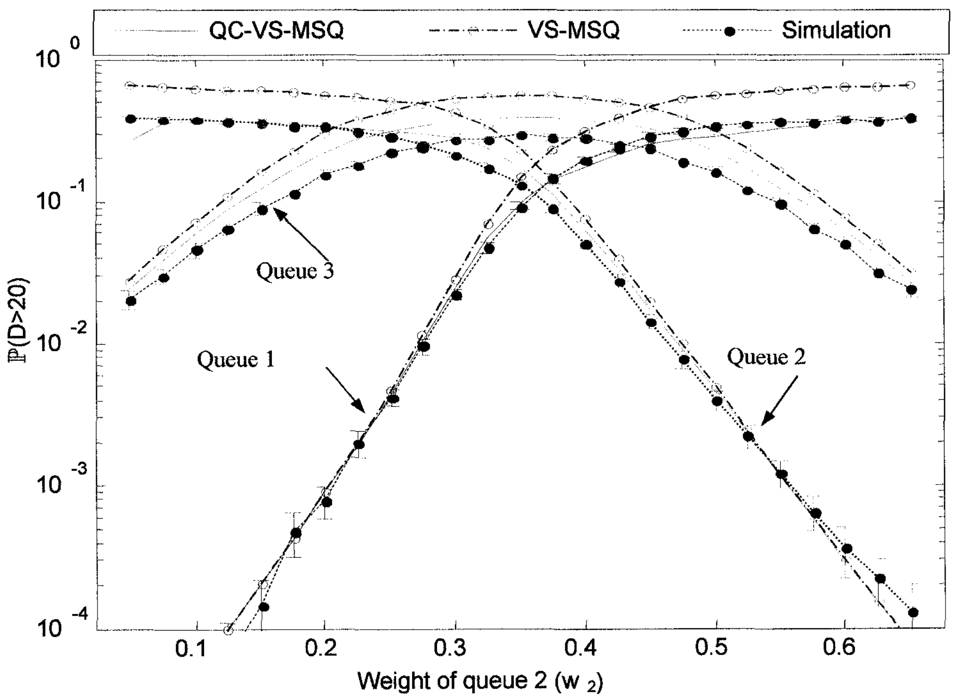


Figure 4-30 $\mathbb{P}(D > 20)$ **versus** w_2 **(with** $w_3 = 0.3$, $w_1 = 0.7 - w_2$**)**

4.6. Chapter Summary

This thesis chapter provides analytical tools that are need to support the traffic engineering architecture presented in Chapter 2. The analysis is necessary for the estimation of the change of delay with load and with the change of weight and priority assignment. This estimation is needed for both the centralized offline optimization and the distributed online problems presented in Chapter 3. The distributed approximations of the optimization problem presented in Chapter 3 depend on the ability of the of each link to use the techniques presented in this chapter to estimate its own delay gradients. This chapter presented an approximation method to calculate the aggregate characteristic of traffic and showed that compared with simulation it presents accurate characterization. To enable multi-queue analysis this chapter presented techniques that are capable of dealing with variable service rate and showed how to analytically estimate the delay survivor function. The core of the analysis is based on multi-level decoupling of the queue in order to obtain the service rate of each decomposed queue. The proposed technique considers all the possible queue states to derive the corresponding service distributions. The analysis provides an insight of the LRD traffic and service behavior of each queue state at a given time scale. Comparison of analytical and simulation results shows that the proposed analytical method provides an accurate estimation of the queue length and delay survivor functions in the presence of variable service rates, and is capable of analyzing the effect of weight variation and service rate characteristics on the queue performance.

Chapter 5

Conclusions and Future Work

This thesis aims to provide an efficient and scalable end-to-end QoS using Internet Protocol backbone networks. It aims to overcome backbone challenges of having to deal with multiple services with varying QoS requirement, using scalable algorithms in order to enable high speed forwarding of large network traffic volumes. This thesis uses DiffServ to provide end-to-end QoS in MPLS networks. The class-based DiffServ provides scalability by grouping connection into a limited number of QoS classes, but lacks the ability to provide accurate end-to-end QoS. Multi-Protocol Label Switching enhances efficiency by introducing Traffic Engineering mechanisms but lacks scalable QoS provisioning. The objective of this thesis is to enable guaranteeing the end-to-end QoS requirements through the proper traffic engineering of class-based networks. Traffic engineering a class-based network faces many challenges. The use of work conserving schedulers to enhance resource sharing introduces dependency between the classes. The dependency requires combining the mapping of network connections to routes and the mapping of routes to QoS classes. Formulating and solving a network optimization problem that combines both is challenging. Implicit reservation of resources requires setting the QoS values to be maintained by each class at each link. Setting QoS values that are suitable for all connections sharing a given class without wasting the link resources is challenging. Implementing these approach in a centralized offline approach enable finding a more optimal network configuration but decreases the ability of these approaches to respond to network changes. Using a distributed online approach is less optimal, but is more capable to adapt to network changes. All these traffic engineering functionalities require the ability to evaluate the performance versus change in traffic characteristics. Recent measurements showed that network traffic displays long-range dependence and self-similarity. These behaviors introduce temporal correlation in the traffic over long periods. The self-similarity of network traffic makes performance evaluation more challenging. In the presence of self-similarity, Markovian models underestimate the queue length survivor function. This underestimation makes traffic-engineering decisions inaccurate. This thesis presents a number of contributions towards overcoming these challenges.

5.1. Contributions

The contribution in this thesis starts by presenting the QoS provisioning approach used to support the proposed delay margin based formulation. The proposed QoS provisioning approach assigns a delay operating point to each link class and allows flows to choose routes that would cause the minimum increase in the network over all delay margins. The approach is enabled by adopting a DiffServ MPLS approach that allows the switching of DiffServ classes with the switching of MPLS labels. The thesis then proposed a traffic engineering architecture that supports the use of the delay margin based approach. The architecture operates on three time scales a long-term network TE, a medium-term flow level TE and a short-term link level TE.

The thesis main contribution is in providing a solution for the class-based traffic engineering optimization problem. The solution is based on non-linear multi-commodity formulation that uses a delay margin penalty function. The penalty function approaches infinity as the delay margin of any connection approaches zero. The form of the penalty functions pushes the TE to choose a network configuration that would keep the delay of end-to-end connections as far way as possible from their requirements. Keeping a large margin between the delay experienced by each end-to-end connection and its requirement has two advantages. It allows the network to accommodate more QoS users by leaving enough delay margins to allow for adding new connections. The margin prevents delay increases that are caused by instantaneous increases in the network traffic. Because the end-to-end delay margin is affected by the connection end-to-end requirement, the route choice, the assignment of the route to delay classes and the implicit reservation of resources in each link, it provides the perfect means to reflecting the inter-class and inter-link dependency introduced by DiffServ and the end-to-end QoS requirements. The formulation dependency allows the use of delay margin to combine the problem of choosing routes for a set of end-to-end connection requirements with the problems of mapping routes to delay classes and defining each class delay operating point

For a single class network, the thesis shows that the function is convex, and introduces a gradient-based technique to solve the problem. The solution is based on decomposing the network delay margin penalty function into link length components. The link lengths represent the rate of increase in delay margin penalty function with the increase in a given link class traffic. Each component captures the effect of all flows using this class and the interacting flows. The thesis shows that each link can separately calculate its own link length component.

In the multi-class case, Chapter 3 shows that concave areas appear in the objective function because of the inter-class dependencies. Chapter 3 shows that these concave areas are the result of splitting the traffic of one end-to-end connection along two or more delay classes on the same link. In Chapter 3 , this thesis presents a multi-class solution that is based on three algorithms. The first two algorithms, *centralized offline route configuration* and *link-class delay assignment*, operate in the *convex* areas of the feasible region to iteratively reduce the objective function using a gradient-based approach. Similar to the single class case both solutions are based on a decomposition of the function gradient into per-link, per-class components. The third algorithm is a heuristic used at the LSP level to promote/demote connections at different links in order to move across the *concave* areas.

The per-link per class problem decomposition is used to provide approximate solutions of the problem that are more suitable for *online distributed* operation. The approximation is based on finding two shortest paths: one that uses delay to represent the link length and the other uses the link length obtained by decomposing the delay margin penalty function. A minimum of the two paths is used to route traffic. The approximation in this form allows for distributed implementation using distance vector routing techniques.

The ability of the centralized offline approach and the distributed online approach to accommodate more network users is tested using simulation. Simulation in this thesis provide results for two types of networks: a network that uses weighted fair queues to schedule traffic at the link level and another that uses the much simpler priority queue. Simulations show that the end-to-end class adjustment and the weight adjustment algorithms can be used separately to enhance the performance of existing routing techniques. In conjunction with the route configuration algorithm, they further increase the network efficiency. When priority queues are used, the approach is further simplified and results show that the proposed approach finds the optimal mapping between QoS requirements and priority levels. Compared with the centralized online approach the offline approximation present a degradation in the number of users that can be supported by the network. However compared with other TE techniques it still presents an increase in the number of satisfied network users.

To enable the proper choice of routing and delay class assignments the thesis contributes an approach to estimate the queue length and delay survivor functions for priority and weighted fair queue schedulers. The technique enables each link to estimate the change of its delay and queue length distribution with the change of its load. The load change results from adding or removing an end-user application to/from a given link. The technique uses two-dimensional decomposition. At first, temporal decomposition uses multi-scale modeling of the traffic and capacity to convert the time-correlated queuing problem into a set of sub-problems over several time scales. Subsequently, queue decomposition exploits the priority or weight dependencies to convert a multi-queue problem into a set of single-queue problems. The core of the analysis is in estimating the multi-scale models for the service available to each of these queues. The thesis shows the hierarchy of this estimation and the dependency of the queue service rate on the unused capacity of other queues and their weights. Comparison with simulation shows the proposed approach can estimate the change in the queue performance versus the change in the traffic characteristics, the capacity characteristics, and versus variation of the queue weights. The analysis also provide means for understanding the components contributing each queue available capacity

5.2. Suggested Future Work

5.2.1. Wireless network cross layer delay margin based traffic engineering

In wireless networks, the delay is affected by the channel characteristic, the channel medium access schemes, as well as the loading and traffic characteristics. Using the delay margin based approach presented in this thesis to traffic engineer wireless networks may present a formulation that combines the effect of all different layers on the QoS received by each connection. By relating delay to power allocation it may be possible to use delay margin as means for combining routing and power allocation in a wireless network.

5.2.2. Multi-Scale Modeling of Wireless Channel Capacity

The existence of multi-path fading causes wireless channel attenuation to vary with time. This variation is found to display temporal correlation. The bit rate that can be supported by channel varies with time as adaptive coding and modulation adapt the bit loading to the channel conditions. This variation affects the QoS received by each connection. The multi-scale approach used in this thesis can be used to model this capacity variation. The capacity multi-scale model can be used to evaluate the QoS of connection using the channel

5.2.3. Adaptive QoS Wireless Network Protocols

The emerging 802.16 employs a class-based wireless channel capacity scheduling and provides means for building mesh networks. It is interesting to investigate if the TE approaches presented in this thesis can be used to traffic engineer QoS in these mesh networks. The evaluation of these protocols can use either simulation platforms or direct implementation on Linux wireless platforms.

References

[1] V. Paxson, S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, volume 3, number 3, June 1995, pp. 226-244.

[2] M.E. Crovella, A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Transactions on Networking*, volume 5, issue 6, December 1997, pp. 835-846,

[3] A. Feldmann, A. C. Gilbert, W. Willinger, T. G. Kurtz, "The Changing Nature of Network Traffic: Scaling Phenomena," *ACM Computer Communication Review*, volume 28, number 2, Apr. 1998, pp. 5-29.

[4] F. P. Kelly, "Notes on effective bandwidths," *in Stochastic Networks: Theory and Applications. ser. Royal Statistical Society Lecture Notes*, F. P. Kelly, S. Zachary, and I. B. Ziedins, Eds. London, U.K.: Oxford Univ. Press, 1996, volume 4, pp. 141–168.

[5] J. R. Gallardo, D. Markrakis, L. Orozco-Barbosa. "Use of alpha-stable self-similar stochastic processes for modeling traffic in broadband networks," *Performance Evaluation,* volume 40, issue 1-3, 2000, pp. 71-98.

[6] A. Kankkunen, "MPLS and next generation access networks," *in Proc. 1st European Conference on Universal Multiservice Networks, ECUMN 2000*, France, 2000, pp.5-16

[7] Hewlett-Packard, "Convergence: Preparing the Enterprise," *White paper* .

[8] D. Thaler, C. Hopps, "Multi-path issues in Unicast and Multicast Next-Hop Selection," *Internet Engineering Task Force Request For Comment,* RFC 2991, November 2001.

[9] R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: an Overview, *Internet Engineering Task Force, Request for Comment,* RFC 1633, June 1994.

[10] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," *ACM Computer Communication Review*, volume 29, issue 4, October 1999, pp. 109-120.

[11] M.K.H. Leung, J.C.S. Lui, D.K.Y. Yau, "Adaptive proportional delay differentiated services: characterization and performance evaluation," *IEEE/ACM Transactions on Networking*, volume 9, issue 6 , December 2001, pp. 801-817.

[12] C. Dovrolis, P. Ramanathan, "A case for relative differentiated services and the proportional differentiation model," *IEEE Network*, volume 13 , issue 5 , September-October. 1999, pp. 26 – 34.

[13] S. Sahu, D. Towsley, J. Kurose, "A quantitative study of differentiated services for the Internet," *in Proc. IEEE Global Internet* , volume 3, Rio de Janeiro, Brazil, 1999, pp. 1808-1817.

[14] [M. May, J. C. Bolot, C. Diot, and A. Jean-Marie, "Simple performance models of differentiated services schemes for the Internet," *in Proc. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 1999*, volume 3, New York, USA, 21-25 March 1999, pp. 1385-1394.

[15] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," *Internet Draft*, November 1997.

[16] F. Kuipers, P. Van Mieghem, T. Korkmaz, M. Krunz, "An overview of constraint-based path selection algorithms for QoS routing," *IEEE Communications Magazine*, , volume 40, issue 12, December 2002 pp. 50-55.

[17] R. Sivakumar, T. Kim, N. Venkitaraman, V. Bharghavan, "Achieving Per-Flow Weighted Rate Fairness in a Core Stateless Network," *in Proc. 20th International Conference on Distributed Computing Systems, ICDCS 2000*, Taipei, Taiwan,10-13 April 2000, pp.188 – 196.

[18] X. Xiao, L. M. Ni, "Internet QoS: a big picture," *IEEE Network Magazine*, volume 13, number 2, March-April 1999, pp. 8–18.

[19] I. Stoica, H. Zhang, "Providing Guaranteed Services Without Per Flow Management," in *Proc. of ACM Special Interest Group on Data Communications Conference, SIGCOMM*, 1999, Cambridge, Massachusetts, United States, August 30-September 03, 1999, pp. 81-94.

[20] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus, "Requirements for Traffic engineering over MPLS," *Internet Engineering Task Force, Request for Comment*, RFC-2702, Sep. 1999.

[21] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, X. Xiao, "Overview and Principles of Internet Traffic Engineering," *Internet Engineering Task Force, Request for Comment,* RFC 3272, May 2002.

[22] H. Kaur, T. Ye, S. Kalyanaraman, K.S. Vastola, "Minimizing packet loss by optimizing OSPF weights using online simulation," *in Proc. IEEE/ACM 11th International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, MASCOTS 2003,* Orlando, Florida, USA, 12-15 October 2003. pp. 79-86.

[23] B. Fortz, J. Rexford, M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Communications Magazine*, volume40, no.10, Oct 2002, pp. 118-124.

[24] B. Fortz, M. Thorup, "Internet traffic engineering by optimizing OSPF weights," *in Proc. IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies., INFOCOM 2000,* volume2, Tele-Aviv, Israel, 26-30 March 2000, pp.519-528.

[25] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, "RSVP-TE: extensions to RSVP for LSP Tunnels," *Internet Engineering Task Force, Request for Comment*, RFC 3209, December 2001.

[26] R. Braden, L. Zhang S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification," *Internet Engineering Task Force, Request for Comment, RFC 2205*, September 1997.

[27] F. Le Faucheur, "Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering," *Internet Engineering Task Force, Request for Comment,* RFC 3564, July 2003.

[28] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, Sandburst, J. Heinanen, T. Kilty, A. Malis,., "Constraint-Based LSP Setup using LDP," *Internet Engineering Task Force, Request for Comment*, RFC 3212, January 2002.

[29] E. Rosen, A. Viswanathan, R. Callon, "Multi-protocol Label Switching Architecture," *Internet Engineering Task Force, Request for Comment,* RFC-3031, Jan. 2001.

[30] D. P. Bertsekas, *Network Optimization Continuous, and Discrete models,* Athena Scientific, 1998.

[31] L. Fratte, M. Gerla , L. Kleinrock, "The flow deviation method: an approach to store and forward Communication network design," *Networks*, volume 2, issue 3, 1973, pp 97-133.

[32] D. P. Bersekas, R. G. Gendron, W. K. Tsai, "Implementation of an optimal Multicommodity Network Flow Algorithm Based on gradient projection and a path flow formulation" Report LIDS-P 1364 Cambridge, MA Laboratory for information and Decision Systems 1984

[33] J.f. Keningtom, " A survey of linear multicommodity network flows ," *Operatinal Research,* volume 2, number 26, 1978, pp 209-236.

[34] J. Chifflet , P. Mahey , V. Renier, "Proximal decomposition for multicommodity network flow problems with convex costs," *Telecommunication Systems,* volume 3, number 1, February 1994, pp 1-10.

[35] H. Nagamochi, "Studies on mulicommodity flows in directed networks" , *Engineering Doctorate Thesis,* Kyoto University, 1988.

[36] J. L Goffin , J. Godzio, R.Sarkissian, J.P Vail, "Decomposition and non-differentialble optimization with the projective algorithm," *Management Science, volume* 38, issue2, 1992, pp 284-302.

[37] R. Nagarajan, J.F. Kurose, D. Towsley, "Allocation of Local Quality of Service Constraints to meet End-to-End Requirements," *in Proc. of International Federation for Information Processing Workshop on the Performance Analysis of ATM Systems,* Martinique, January 1993.

[38] V. Firoiu, D. Towsley, "Call admission and resource reservation for multicast sessions," *in Proc. IEEE Fifteenth Annual Joint Conference of the IEEE Computer Societies, INFOCOM '96,* volume 1, San Francisco, CA, USA, 24-28 March 1996, pp. 94 – 101.

[39] D.H. Lorenz, A. Orda, "QoS routing in networks with uncertain parameters," *IEEE/ACM Transactions on Networking*, volume 6 , issue 6 , Dec. 1998, pp. 768–778.

[40] G. Lapiotis, S. Mao, S. Panwar, "GPS Analysis of Multiple Sessions with Applications to Admission Control ," *in Proc. IEEE International Conference on Communications, ICC 2001*, volume 6, Helsinki, Finland, 11-14 June 2001, pp 1829 - 1833.

[41] A. Elwalid, D. Mitra, "Traffic shaping at a network node: theory, optimum design, admission control," *in Proc. Sixteenth Annual Joint Conference of the IEEE Computer*

and Communications Societies, INFOCOM '97, volume 2 , Kobe, Japan, 7-11 April 1997, pp. 444 – 454.

[42] H. Inai, "End-to-end rate-based flow control over ATM Networks," 1993. *Proc. of IEEE Singapore International Conference Information Engineering* '93. volume 1 , 6-11 Sept. 1993 pp. 460 – 464.

[43] A. Orda, "Routing with end-to-end QoS guarantees in broadband networks," *IEEE/ACM Transactions on Networking,* volume 7 , issue 3 , June 1999, pp. 365 – 374.

[44] D. Yates, J. Kurose, D. Towsley, M. Hluchyj, "On Per-Session End-To-End Delay and the Call Admission Problem for Real Time Applications with QoS Requirements," *Journal of High Speed Networks,* volume 3, December 1994, pp. 429-458.

[45] G. Apostolopoulos, S. K. Tripathi, "On the Effectiveness of Path Pre-computation in Reducing the Processing Cost of On-demand QoS Path Computation," *in Proc. IEEE Symposium on Computers and Communication*, Athens, Greece, 30 June-2 July 1998, pp. 42-46.

[46] C. Partridge, "A Proposed Flow Specification," *Internet Engineering Task Force Request for Comments,* RFC-1363, 1992.

[47] D. Lorenz, A. Orda, "Optimal partition of QoS requirements on unicast paths and multicast trees," *IEEE/ACM Transactions on Networking,* volume 10 , issue 1 , February 2002 pp. 102 – 114.

[48] R. Nagarajan, J. Kurose, "On Defining, Computing, and Guaranteeing Quality-of-Service in High-Speed Networks," *Proc. of Annual joint conference of the IEEE Computer and Communications Societies, INFOCOM'92*, Florence, Italy,4-8 May 1992, pp. 2016-2025.

[49] M. Ashour, T. Le-Ngoc, "End-to-end delay margin balancing approach for routing in multi-class networks," *Wireless Network* , volume 13, issue3, June 2007, pp. 311-322.

[50] M. Ashour, T. Le-Ngoc, "End-to-End Delay Satisfaction Balanced Routing," *in Proc. IEEE Global Telecommunications Conference, GLOBECOM 2005,* volume 2, St. Louis, Missouri, USA, 28 Novmber-2 December 2005, pp. 852-856.

[51] M Ashour, B. Kassem, T. Le-Ngoc, T. El-Shabrawy, "A path tuning algorithm for short-term traffic engineering of QoS differentiated MPLS networks," *in Proc. IEEE Canadian Conference on Electrical and Computer Engineering CCECE 2003,* volume 2, Montreal, Canada, 4-7 May 2003 volume2, pp. 839-842.

[52] M. Ashour, T. Le-Ngoc, "Priority Queuing of Long Range Dependent Traffic," *in Proc. IEEE Global Communications Conference, GLOBECOM 2003*, San Francisco, volume 6, 1-5 December 2003, pp. 3025- 3029.

[53] M. Ashour, T. Le-Ngoc, "Multi-Scale Queuing Analysis of Long Range Dependent Traffic and Variable Service Rates," *in Proc. IEEE International Conference on Communications, ICC2006,* volume 2, Istanbul, Turkey. 11-15 June 2006, pp. 585 – 590.

[54] M. Ashour, T. Le-Ngoc "Performance of Weighted Fair Queuing Systems with Long Range Dependent Traffic Inputs," *in Proc. IEEE Canadian Conference on Electrical and Computer Engineering, IEEE CCEC05*, Saskatoon, Saskatchewan Canada, May 1-4, 2005 pp. 1002 – 1005.

[55] M Ashour, T. Le-Ngoc, "Aggregation of long-range dependent traffic streams using Multi-fractal wavelet models," *in Proc. IEEE Canadian Conference on Electrical and Computer Engineering, CCECE 2003*, volume 2, Montreal, Canada, 4-7 May 2003, Page(s):793 – 796

[56] M. Ashour, T. Le-Ngoc, "Performance Analysis of Weighted Fair Queues with Variable Service Rates," *in Proc International Conference on Digital Telecommunications, ICDT 2006,* Côte d'Azur, France, 30-31 August 2006, pp.51-56.

[57] S. Ma, C. Ji, "Modeling video traffic in the wavelet domain," *17th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 98*, volume 1, San Francisco, USA, 29 March-2 April 1998, pp. 201–208.

[58] I. Hu, M. d Ashour, T. Ngoc, Y. Lemieux, "Traffic Engineering Evaluation Platform for QoS-Aware DiffServ-MPLS Networks," *in Proc. IEEE Canadian Conference on Electrical and Computer Engineering CCECE 2005*, Saskatoon, Saskatchewan Canada, May 1-4, pp. 1788-1791.

[59] Y. Lemieux, M. Ashour, T. Elshabrawy, "Quality of service (QoS) mechanism in an internet protocol (IP) network," United States Patent No: US 6,968,374 B2, issued: 22 November 2005.

[60] L. Kleinrock, *Queuing Systems: volume I – Theory*, Wiley Interscience, New York, 1975.

[61] M. Kodialam, T.V. Lakshman, " Minimum interference routing with application to MPLS traffic engineering," *in Proc. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2000,* Tele-Aviv, Israel, 26-30 March 2000, pp. 884-893.

[62] H. Heffes, D. M. Lucantoni, "A Markov Modulated characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," *IEEE Journal of Selected Areas Communication,* volume 4, issue 6, September 1986, pp.856-868.

[63] P. Aukia, M. Kodialam, P.V.N. Koppol, T.V. Lakshman,; H. Sarin, B. Suter, "RATES: a server for MPLS traffic engineering," *IEEE Network*, volume 14 , issue: 2 , March-April 2000 pp. 3-41

[64] T. S. Choi, S. H. Yoon, H. S. Chung, C. H. Kim, J. S. Park, B. J. Lee, T. S. Jeong, "Wise: traffic engineering server for a large-scale MPLS-based IP network," *in Proc. IEEE/IFIP Network Operations and Management Symposium, NOMS 2002*, Florence, Italy,15-19 April 2002 pp. 251-264.

[65] WANDL www.WANDL.com

[66] K. Matsui, M. Kaneda, H. Takenaka, H. Ishii, M. Imase, "TRES: traffic and reliability engineering server for MPLS," *in Proc IEEE Pacific Rim Conference on Communications, Computers and signal Processing,* PACRIM 2001, volume 2, Victoria, B.C., Canada, 26-28 Aug. 2001, pp. 583 – 586.

[67] P Nanda, A Simmonds: "Providing end-to-end guaranteed quality of service over the Internet: a survey on bandwidth broker architecture for differentiated services network', *in Proc. 4th International Conference on IT, CIT 2001,* Berhampur, India, 20–23 December 2001, pp. 211–216.

[68] S. Blake, D. Black, M. Carlson, E. Davies, Z Wang, W. Weiss, "An Architecture for Differentiated Services," *Internet Engineering Task Force, Request for Comment*, RFC-2475, Dec. 1998.

[69] S. Bakiras, V.O.K. Li, "Quality of service support in differentiated services packet networks," *in Proc. IEEE International Conference on Communications, ICC 2001,* volume 8 , Helsinki, Finland, 11-14 June 2001, pp. 2370-2374.

[70] S.K. Biswas, S. Ganguly, R. Izmailov, "Path provisioning for service level agreements in Differentiated Services networks," *in proc. IEEE International Conference on Communications, ICC 2002* , volume 2 , New York, USA, 28 April-2 May 2002 pp. 1063 – 1068.

[71] K. Matsui, J Watase, M. Kaneda, N. Tanaka, H. Ichikawa, "A routing method with traffic balancing over connectionless network," *in Proc. Institute of electronics information and communication Spring Conf,* March 2000.

[72] C. Dovrolis, P. Ramanathan, "Dynamic class selection: From relative differentiation to absolute QoS," *in Proc. Ninth International Conference on Network Protocols, ICPN 2001*, 11-14 Nov. 2001, pp. 120-128.

[73] W. Fugui; P. Mohapatra, S. Mukherjee, D. Bushmitch, "A random early demotion and promotion marker for assured services," *IEEE Journal on Selected Areas in Communications,* volume 18 , issue. 12 , December 2000, pp. 2640 - 2650

[74] S. Chen, K. Park, "An architecture for non-cooperative QoS provision in many-switch systems," *in Proc IEEE Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies., INFOCOM '99* , volume 2 , New York, NY, USA, 21-25 March 1999, pp. 864 – 872.

[75] H. Wang, C. Shen, K. G. Shin, "Adaptive-weighted packet scheduling for premium service," *in Proc. IEEE International Conference on Communications, ICC 2001,* volume 6, Helsinki, Finland, 11-14 June 2001, pp.1846 - 1850.

[76] R. Liao, A. T. Campbell, "Dynamic core provisioning for quantitative differentiated services," *IEEE/ACM Transactions on Networking* , volume12, no.3, June 2004, pp. 429-442.

[77] M. Homg, W. Lee, K. Lee, Y. Kuo, "An adaptive approach to weighted fair queue with QoS enhanced on IP network," *in Proc. IEEE Region 10 International Conference on Electrical and Electronic Technology, TENCON 2001*, volume1, Singapore, 19- 22 August, 2001 pp.181-186.

[78] L. Chin-Chang, T. Shiao-Li, C. C. Meng, S. Yeali, H. Yueh-Min, "Proportional delay differentiation service-based on weighted fair queuing," *in Proc. Ninth International Conference on Computer Communications and Networks, ICCCN 2000,* Las Vegas, Nevada, USA, 16 – 18 October 2000, pp.418-423.

[79] D. Hang, H. Shao, W. Zhu, Y. Zhang, "TD^2FQ: an integrated traffic scheduling and shaping scheme for DiffServ networks," *in Proc. IEEE Workshop on High Performance Switching and Routing,* Dallas, TX,USA, 29-31 May 2001, pp.78-82.

[80] R. K. Ahja, T.L Magnanti, J. Orlin, *Network Flows: Theory, Algorithms And Application*, Printce Hall.

[81] S. P. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press (Chapter 11) 2004.

[82] B. Gavish, I. Neuman, "Routing and capacity assignment in Computer communication Networks," *IEEE Transaction on communication,* volume 37, number 4, April 1989, pp. 360-366.

[83] M. Geral, L. Kelinrock, "On the topological design of distributed computer networks," *IEEE Transactions on Communications,* volume25, number 1, Jan 1977 , pp. 48-60.

[84] A. Girard, R. Zidane, "Revenue optimization of B-ISDN networks," *IEEE Transactions on communications,* volume 43 , issue 5, May 1995, pp. 1992–1997.

[85] R. Guerin, A. Orda, "QoS based routing in networks with inaccurate information: theory and algorithms," *in Proc. IEEE Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '97*, volume 1, Kobe, Japan, 7-11 April 1997 pp. 75 – 83.

[86] L. Atov, H. T. Tran; R. J. Harris, "Efficient QoS partition and routing in multiservice IP networks," *in Proc. IEEE International Performance, Computing, and Communications Conference,* IPCCC 2003, Phoenix, Arizona, 9-11 April 2003, pp. 435 – 441.

[87] M. Moh, B. Wei, J.H. Zhu, "Supporting differentiated services with per-class traffic engineering in MPLS," *in Proc. of 2001 Tenth International Conference on Computer Communications and Networks," ICCCN 2001,* Scottsdale, Arizona, USA,15-17 Oct. 2001, pp. 354–360.

[88] D. Bienstock , O. Raskina, "Asymptotic Analysis of the Flow Deviation Method for the Maximum Concurrent Flow Problem" , *Math. Programming*, volume 91, 2002, pp.479 – 492.

[89] D. H. Lorenz, A. Orda, "Optimal portioning of QoS requirements on Unicast and Multicast networks," *IEEE/ACM Transactions on Networking,* volume 10, no. 1, Feb. 2002, pp. 102 – 114.

[90] T. Ibaraki , N. Katoh, *Resource allocation problems. The foundation of computing.* MIT press, Cambridge, April 1998

[91] Dorit S. Hochbaum, Lower and Upper Bounds for the Allocation Problem and Other Nonlinear Optimization Problems," *Mathematics of Operations Research*, volume 19, number 2, May 1994, pp. 390-409.

[92] J. F. Hayes, T. G. Babu, *Modeling and Analysis of Telecommunications Networks*, (John Wiley & Sons, 2004).

[93] K. Kar, M. Kodialam, T. V. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *IEEE Journal of Selected Areas in Communication,* volume 18, number. 12 , December 2000, pp. 2566-2579.

[94] A. Orda, A. Sprintson," QoS routing: the precomputation perspective," *in Proc. IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2000,* volume 1 , Tele-Aviv, Israel, 26-30 March 2000, pp. 128-136.

[95] V. J. Ribeiro, R. H. Riedi, M. S. Crouse, R. G. Baraniuk, "Multiscale Queuing Analysis of Long-Range-Dependent Network Traffic," *in Proc. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2000,* March 26 - 30, 2000, Tel-Aviv, Israel, volume 2, pp. 1026 – 1035.

[96] A. Khamisy, M. Sidi, "Discrete-time priority queues with two-state Markov Modulated arrivals," *Stochastic Models, Vol* 8, issue 2, 1992, pp. 337-357.

[97] J. Zhang, "Performance study of Markov modulated fluid flow models with priority traffic," *in Proc. Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM* '93, volume 1, 28 March-1 April 1993, pp10-17.

[98] F. Bonomi, J. Meyer, S. Montagna,, R. Paglino, "Minimal on/off source models for ATM traffic," *Proc. ITC14,* Eds. J. Labetoulle and J. W. Roberts, Elsevier, pp287-400.

[99] T. Ferrari, G. Pau, C. Raffaelli, "Measurement Based Analysis of Delay in Priority Queuing," *in Proc. IEEE Global Telecommunication Conference, GLOBECOM 2001,* volume 3, S. Antonio, Texas, Nov 25-29 2001, pp.1834-1840.

[100] B. Venkataramani, Sanjay K. Bose, K.R. Srivathsan, "Queuing analysis of a non-preemptive MMPP/D/1 priority system," *Computer Communications Journal*, volume 20, issue 11, 15-October 1997, pp 999-1018

[101] Elwalid, D. Mitra, "Analysis, Approximations and Admission Control of a Multi-Service Multiplexing System with Priorities," *in Proc. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '95,* Boston, MA, USA, 2-6 Apr 1995, pp. 463-472.

[102] A. W. Berger, W. Whitt, "Workload bounds in fluid models with priorities," *Perform. Evaluation.,* volume 41, issue 4, August 2000, pp. 249-267.

[103] A.K. Parekh, R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking*, volume 1, issue 3, June 1993, pp. 344 –357.

[104] Z.-L. Zhang, *End-to-End Support for Statistical Quality-of-Service Guarantees in Multimedia Networks*, Ph.D Dissertation, Department of Computer Science, University of Massachusetts, February 1997.

[105] O. Yaron, M. Sidi, "Exponentially bounded Generalized processor sharing networks with exponentially bounded burstiness arrivals," *in Proc. Thirteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 94,* volume 2 , *Toronto, Ontario, Canada, 12-16 June 1994,* pp. 628 – 634.

[106] X. Yu, I.L.-J. Thng, Y. Jiang, C. Qiao, "Queuing Processes in GPS and PGPS With LRD Traffic Inputs," *IEEE/ACM Transactions on Networking*, volume 13, issue 3, June 2005, pp. 676 – 689 .

[107] F. Pererira, N. Fonseca, D. Arantes, "On the performance of generalized processor sharing server under long range dependent traffic," *Computer Networks: The International Journal of Computer and Telecommunications Networking, Special issue: Advances in modeling and engineering of Longe-Range dependent traffic*, volume 40 , issue 3, October 2002, pp. 413 – 431.

[108] F.L. Presti , Z. Zhang, D. Towsley, "Bounds, approximations and applications for a two-queue GPS system," *in Proc. Fifteenth Annual Joint Conference of the IEEE Computer Societies, IEEE INFOCOM '96,* volume 3, San Francisco, USA, 24-28 March 1996, pp. 1310 – 1317.

[109] R. Agrawal , Makowski, P. Nain "On a reduced load equivalence for fluid queues under sup-exponentially," *Queuing Systems: Theory and Applications* , volume 33, issue 1-3, March 1999, pp. 5 – 41.

[110] S. Borst, M. Mandjes, M. V. Uitert, "Generalized processor sharing with light-tailed and heavy-tailed input," *IEEE/ACM Transactions on Networking*, volume11, issue 5, October 2003, pp. 821 – 834.

[111] S. Borst, O. Boxma, P. Jelenkovic, "Reduced-Load Equivalence and Induced Burstiness in GPS Queues with Long-Tailed Traffic Flows," *Queueing Systems*, volume 43, issue. 4, April 2003, pp. 273 – 306.

[112] G. Lapiotis, S. Panwar, "Quality of service analysis of shared buffer management policies combined with generalized processor sharing," *in Proc. IEEE Global*

Telecommunications Conference, GLOBECOM '99 , volume 1, Rio de Janeiro, Brazil, 05- 09 December 1999 pp. 37 – 43.

[113] P. Mannersalo, I. Norros. "GPS schedulers and Gaussian traffic," *Proc. 21st Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2002,* volume 3, New York , USA, 23-27 June 2002, pp. 1660–1667.

[114] V.J. Ribeiro, R.H. Riedi,, R.G. Baraniuk, "Multiscale Queuing Analysis," *IEEE/ACM Transactions on Networking*, volume14, number 5, Oct. 2006, pp. 1005-1018.

[115] W. E. Leland, M. S. Taqqu, W. Willinger, , and D. V. Wilson, "On the self-similar nature of Ethernet traffic," *IEEE/ACM Transaction on Network,* volume 2, No. 1, Feb. 1994, pp 1-15.

[116] TR 101 112 V3.2.0 (1998-04), "Universal Mobile Telecommunications System (UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS," Annex B.

[117] C. R. Bough, J. Huang., "Traffic Model for 802.16 TG3 MAC/PHY Simulations," IEEE 802.16.3c-01/30r1, March, 2001.

[118] M. Montgomery, G. De Veciana, "On the relevance of time scales in performance oriented traffic characterizations" *in Proc. Fifteenth Annual Joint Conference of the IEEE Computer Societies, INFOCOM '96,* volume 2 , San Francisco, CA, USA, 24-28 March 1996, pp. 513 -520.

[119] R. H. Riedi, M. S. Crouse, V. J. Ribeiro, R. G. Baraniuk, "A Multifractal Wavelet Model with Application to Network Traffic," *IEEE Transactions on Information Theory*, Vol 45, No.3, April 1999, pp. 992-1018.

[120] N.L. Johnson, S. Kotz, N. Balakrishnan, *Continuous Univariate Distributions,* volume 2, 2nd edition. Wiley, New York. 1994.