QoS Provision and Routing with Deterministic and Stochastic Guarantees

Erez Biton
QoS Provision and Routing with Deterministic and Stochastic Guarantees

Research Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Erez Biton

Submitted to the Senate of the Technion - Israel Institute of Technology
Sivan 5765 Haifa June 2005
This research thesis was done under the supervision of Prof. Ariel Orda in the Department of Electrical Engineering.

The generous financial help of the Technion and the Committee for Planning and Budgeting at the Ministry of Science are gratefully acknowledged.

I am deeply indebted to my advisor, Professor Ariel Orda, for his professional guidance, for his personal support, and for all that he has taught me.

I would like to thank several colleagues for stimulating discussions and fruitful collaborations, especially Ron Banner, Guy Gilboa, Omer Gurewitz, Doron Karni, Nadav Lavi, Lavi Libman, Dean Lorenz, Nir Naaman, and Gil Zussman.

I would like to acknowledge the support by Yoram Or-Chen and the Computer Networks laboratory staff.

I would also like to thank Lucent Technologies, and particularly Dr. David Lee, for hosting me for an internship in the summer of 2002.

Finally, I am deeply grateful to my parents, to whom I dedicate this thesis.
# Contents

Abstract 1

Notations and abbreviations 2

1 Introduction 5

1.1 Background and Related Work . . . . . . . . . . . . . . . . . . . . . . 7
    1.1.1 Traffic Specifications . . . . . . . . . . . . . . . . . . . . . . . 7
    1.1.2 Scheduling Policies . . . . . . . . . . . . . . . . . . . . . . . . 9
    1.1.3 End-to-End Guarantees . . . . . . . . . . . . . . . . . . . . . . 11
    1.1.4 Routing Algorithms . . . . . . . . . . . . . . . . . . . . . . . . 13
1.2 Overview of Major Results . . . . . . . . . . . . . . . . . . . . . . . . 15

2 The GPS scheduling discipline 17

2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
2.2 Model Formulation . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
2.3 Deterministic Setting . . . . . . . . . . . . . . . . . . . . . . . . . . 21
    2.3.1 Rate Consumption Criterion . . . . . . . . . . . . . . . . . . . . 22
    2.3.2 Load Balancing Criterion . . . . . . . . . . . . . . . . . . . . . 25
    2.3.3 Variable-Rate Links . . . . . . . . . . . . . . . . . . . . . . . . 27
2.4 Stochastic setting . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
    2.4.1 Stochastic Guarantees - EBB Traffic . . . . . . . . . . . . . . 29
    2.4.2 Simulation results . . . . . . . . . . . . . . . . . . . . . . . . . 37
    2.4.3 Stochastic Guarantees - SBB Traffic . . . . . . . . . . . . . . 40
    2.4.4 Variable-Rate Links . . . . . . . . . . . . . . . . . . . . . . . . 44
2.5 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
2.6 Appendix . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
    2.6.1 Proof of Proposition 2.4 . . . . . . . . . . . . . . . . . . . . . 50
2.6.2 Proof of Proposition 2.6 ........................................... 51
2.6.3 Proof of Proposition 2.8 ........................................... 54
2.6.4 Proof of Proposition 2.9 ........................................... 55
2.6.5 Proof of Proposition 2.11 ......................................... 57
2.6.6 Proof of Proposition 2.12 ......................................... 58
2.6.7 Proof of Proposition 2.14 ......................................... 59

3 The EDF scheduling discipline - deterministic setting 62
3.1 Introduction ......................................................... 62
3.2 Model Formulation ............................................... 63
3.3 EDF Schedulability Conditions ............................... 64
3.4 Finding Feasible Paths ........................................... 66
  3.4.1 Reshaping both the traffic burst as well as the traffic rate . 68
  3.4.2 Reshaping only the traffic burst ............................ 73
3.5 Optimizing the Path Selection .................................. 80
  3.5.1 Rate Consumption Criterion ................................. 80
  3.5.2 Delay-Rate Consumption Criterion ......................... 82
3.6 Simulation Results ............................................... 86
3.7 Conclusion .......................................................... 89

4 The EDF scheduling discipline - stochastic setting 91
4.1 Introduction ......................................................... 91
4.2 Model Formulation ............................................... 92
4.3 The single node case ............................................. 93
  4.3.1 EDF Scheduler ................................................ 93
  4.3.2 EBB traffic shaper .......................................... 98
4.4 The multiple node case ......................................... 101
  4.4.1 Traffic shaper elements in series .......................... 101
  4.4.2 EDF-Scheduler and EBB-Shaper in Cascade ............... 101
  4.4.3 End-to-end Delay ........................................... 103
  4.4.4 Achieving RPPS GPS Guarantees ......................... 104
4.5 Conclusion .......................................................... 105
4.6 Appendix ............................................................ 106
5 Call Admission Control in Fault-Tolerant Networks
  5.1 Introduction ..................................................... 113
  5.2 Optical networks with shared protection ...................... 115
  5.3 Wireless LAN networks ........................................... 123
  5.4 Numerical Examples ............................................... 127
  5.5 Conclusion ....................................................... 129

6 Conclusion
  6.1 Non-worst-case analysis ......................................... 131
  6.2 Hierarchical networks and topology aggregation ............. 132

Bibliography 134
List of Figures

2.1 Algorithm Minimum Rate - Rate Quantized (MR-RQ) ............... 22
2.2 Algorithm Minimum Relative Rate - Rate Quantized (MRR-RQ) .... 25
2.3 Algorithm Minimum Delay tail Distribution (MDD) ............... 33
2.4 Algorithm Minimum Cost (MC) .................................. 34
2.5 Algorithm Quickest Path - Rate Quantized (QP-RQ) ............. 36
2.6 A simple network example ....................................... 37
2.7 Bounds and actual Bernoulli traffic ............................... 38
2.8 Networks topologies .............................................. 38
2.9 Flow request blocking probability - evenly distributed load .... 39
2.10 Flow request blocking probability - unevenly distributed load ... 40
2.11 An example network ............................................. 43
2.12 SBB network example ( - Simulation; — SB bound; ··· EB bound) 44
2.13 Algorithm Minimum Delay tail Distribution - Rate Quantized (MDD-RQ) ........................................... 45
2.14 Algorithm Minimum Delay tail Distribution - Approximation (MDD-A) 48
3.1 A typical instance of the work availability function ............... 65
3.2 Algorithm Minimum-Link-Delay .................................... 69
3.3 Algorithm QFPTS .................................................. 70
3.4 Algorithm QFPTS-Q-2 ............................................. 71
3.5 A 6-hop network example ......................................... 73
3.6 A typical instance of the link-delay-burst function $W_l(\sigma^n_l)$ ... 74
3.7 Algorithm Calculate-Link-Delay-Burst ............................ 75
3.8 Algorithm QFPTS-FR .............................................. 78
3.9 Algorithm QFPTS-FR-RQ-$\epsilon$ ................................ 79
3.10 Algorithm MRB .................................................... 81
3.11 A typical instance of the residual resource function .............. 82
3.12 Algorithm Calculate-Residual-Resource .......................... 84
3.13 Typical instances of the residual resource function ............... 85
3.14 Algorithm MRR ...................................................... 87
3.15 Algorithm MRR (sketch) ............................................ 88
3.16 Network topologies .................................................. 88
3.17 Session request blocking probability ............................... 89
4.1 EDF vs. GPS schedulable region .................................... 96
4.2 The Systems \( S_1 \) and \( S_2 \) ........................................ 102
4.3 Original and Modified Systems ...................................... 103
5.1 Symbolic representation of the state transition structure for the admission control model ................................. 116
5.2 Uniformization of the admission control model ..................... 117
5.3 Example 5.1 ......................................................... 128
5.4 Example 5.2 ......................................................... 128
List of Tables

2.1 Sessions parameters ........................................ 38
2.2 Arrival processes parameters ............................. 43
Abstract

End-to-end Quality-of-Service (QoS) provision and routing are central and critical issues in the design of integrated multimedia networks. Indeed, end-to-end support for QoS has been widely investigated. In particular, scheduling disciplines for guaranteed performance service as well as worst-case end-to-end performance bounds have been established and explored in a large number of studies, under both deterministic and stochastic settings. Recently, the corresponding routing problems have been addressed as well. However, a comprehensive study, which considers both problems, i.e., the establishment of end-to-end performance bounds and the corresponding routing problems, has not been reported. Since these two problems are closely related, moreover the former is a prerequisite for the latter, in this research they are investigated as a whole. In other words, this work presents a methodology for providing QoS guarantees by considering the coupling between the scheduling mechanism and the routing schemes. More specifically, we consider QoS provision and routing schemes for connections with end-to-end delay requirements in networks that employ rate-based schedulers.

In the first part of the dissertation, we focus on the generalized processor sharing (GPS) scheduling discipline. Here, we study three settings: (i) burstiness-constrained (BC) traffic with deterministic QoS requirements, (ii) exponentially bounded burstiness (EBB) traffic with stochastic QoS requirements, and (iii) (general) stochastic bounded burstiness (SBB) traffic. Then, we turn our attention to the rate-controlled earliest deadline first (EDF) scheduling discipline. Here, we consider BC as well as EBB traffic. For each, we obtain end-to-end delay bounds for packetized traffic and links with non-negligible propagation delays, and, consequently, we formulate appropriate routing schemes that identify feasible paths under several network optimization criteria.

Finally, we consider the provision of QoS in fault tolerant networks. Here, the required QoS is given in terms of call termination probability upon resource failures. We consider the optimal admission control policies for two network models, namely, optical networks with shared protection and wireless networks.
### Notations and abbreviations

\[ G(V, E) \] directed graph
\[ V \] set of nodes
\[ E \] set of links
\[ N \] number of nodes in \( G(V, E) \)
\[ M \] number of links in \( G(V, E) \)
\[ l \] link \( l \in E \)
\[ R_l \] service rate of link \( l \)
\[ d_l \] constant delay at link \( l \)
\[ r_i \] session \( i \) assigned rate
\[ R'_l \] the maximal available rate that link \( l \) can offer to a new session
\[ K \] the number of maximal rate values in the network
\[ g^i_l \] session \( i \) guaranteed rate at link \( l \)
\[ L^i \] session \( i \) maximal packet size
\[ L_{\text{max}} \] all sessions maximal packet size
\[ D^i \] session \( i \) required end-to-end delay
\[ q^i \] session \( i \) “sensitivity” to end-to-end delay fluctuations
\[ p \] path
\[ H \] maximal number of hops along a path
\[ n(p) \] number of hops (links) along a path \( p \)
\[ r(p) \] maximal available rate along a path \( p \)
\[ D^i(p, r) \] session \( i \) end-to-end delay along a path \( p \) with a reserved rate \( r \)
\[ D^i(p) \] minimum value of \( D^i(p, r) \)
\[ g^i_p \] session \( i \) guaranteed service rate along a path \( p \)
\[ I(l) \] set of sessions present at link \( l \)
\[ A(t) \] traffic rate at time \( t \)
\( (\sigma, \rho) \) BC traffic parameters
\( \sigma \) BC traffic maximal burst size
\( \rho \) traffic long term upper rate
\( (\rho, \Lambda, \alpha) \) EBB traffic parameters
\( \Lambda \) linear parameter of an EBB traffic
\( \alpha \) decay parameter of an EBB traffic

---

1Notations are ordered according to their appearance; abbreviations are ordered alphabetically
\( A^i (\tau, t) \) the amount of session \( i \) traffic that arrive in an interval \( (\tau, t] \)

\( S^i (\tau, t) \) the amount of session \( i \) traffic served in an interval \( (\tau, t] \)

\( A^i_l \) session \( i \) arrival process into server \( l \)

\( S^i_l \) session \( i \) departure process from server \( l \)

\( A^i_l(n) \) session \( i \) arrival process into the \( n \)-th hop

\( S^i_l(n) \) session \( i \) departure process at the \( n \)-th hop

\( A^i_{(n)} \) session \( i \) arrival process into the first server on its path

\( S^i_{(n)} \) session \( i \) departure process at the \( n \)-th hop

\( Q^i_l(t) \) session \( i \) backlog at the \( n \)-th hop at time \( t \)

\( Q^i_p(t) \) session \( i \) total backlog over a path \( p \) at time \( t \)

\( D^i_p(t) \) session \( i \) instantaneous end-to-end delay over a path \( p \) at time \( t \)

\( \bar{D}^i(p, q^i) \) session \( i \) effective end-to-end delay at a path \( p \)

\( (\Delta, r) \) FC server parameters

\( \Delta \) maximal rate fluctuation in a FC server

\( (r, B, \beta) \) EBF server parameters

\( B \) linear parameter of an EBF server

\( \beta \) decay parameter of an EBF server

\( \delta^i_l \) session \( i \) deadline at an EDF server at link \( l \)

\( F_l(t) \) link \( l \) work availability function

\( (\delta^i_l, w^i_l, r^i_l)_{0 \leq i \leq l_i} \) set of deadlines, minima and slopes that specify the link \( l \) work availability function

\( \delta^m_{l(0)} \) minimum deadline that can be guaranteed to a session \( m \) at link \( l \)

\( \delta^m \) set of deadline assignments to session \( m \) at each link \( l \in p \)

\( (\sigma^m_l, \rho^m_l) \) session \( m \) reshaping parameters at link \( l \)

\( (\sigma^m_0, \rho^m_0) \) session \( m \) reshaping parameters at the entrance to the network

\( (\sigma^m, \rho^m) \) set of reshaping parameters of session \( m \) at each link \( l \in p \)

\( W_l(\sigma^m_l, \rho^m_l) \) minimum delay that can be guaranteed to a session \( m \) at link \( l \) as a function of the reshaping parameters \( (\sigma^m_l, \rho^m_l) \)

\( W_l(\sigma^m_l) \) minimum delay that can be guaranteed to a session \( m \) at link \( l \) as a function of the reshaping parameter \( \sigma^m_l \) with \( \rho^m_l = \rho^m_0 \)

\( S_l(d) \) link \( l \) residual resource function

\( \gamma^j \) session \( j \) guaranteed effective delay with respect to \( q^j \) at an EDF scheduler

\( D_i(A_0, A_1) \) the delay suffered by the \( i \)-th packet of traffic \( A_0 \) at a traffic shaper \( A_1 \)

\( W_{\rho}(A)(t) \) the backlog at time \( t \) at a work-conserving system with an input traffic \( A \) and a transmit rate \( \rho \)
\( \lambda \) connections arrival rate
\( \mu \) connections service rate
\( \sigma \) resources failure rate
\( \tau \) resources repair rate
\( r \) connection acceptance reward
\( p \) premature connection termination penalty
\( (x, m) \) Markov process state parameters
\( x \) connections
\( m \) resources
\( V_n(x, m) \) MDP value function at the \( n \)-th value iteration
\( T_\ast \) dynamic programming event operator
\( T_A \) arrival operator
\( T_{MD} \) multi-departure operator
\( T_F \) failure operator
\( T_{MF} \) multi-failure operator
\( T_R \) repair operator
\( T_{MR} \) multi-repair operator
\( C \) cost operator

BC Burstiness Constrained
CAC Call Admission Control
CRST Consistent Relative Session Treatment
CSMA/CA Carrier Sense Multiple Access - Collision Avoidance
EB Exponentially Bounded
EBB Exponentially Bounded Burstiness
EBF Exponentially Bounded Fluctuation
EDF Earliest Deadline First
FC Fluctuation Constrained
FCFS First Come First Served
FFQ Fluid Fair Queuing
GPS Generalized Processor Sharing
LBAP Linear Bounded Arrival Process
MAC Media Access Control
MDP Markov Decision Process
MWRT Minimum Worst-case Response Time
NPEDF Non-Preemptive Earliest Deadline First
OXC Optical Cross Connect
PGPS Packetized Generalized Processor Sharing
QoS Quality of Service
RC-EDF Rate-Controlled Earliest Deadline First
RPPS Rate Proportional Processor Sharing
RSP Restricted Shortest Path
SB Stochastically Bounded
SBB Stochastically Bounded Burstiness
SCFQ Self-Clocked Fair Queuing
STFQ Start-Time Fair Queuing
WFQ Weighted Fair Queuing
WLAN Wireless Local Area Network
Chapter 1

Introduction

End-to-end *Quality-of-Service (QoS)* provision and routing are central and critical issues in the design of integrated multimedia networks. A QoS architecture should provide applications with stringent end-to-end guarantees such as bandwidth, delay, and packet loss. The provision of QoS involves a broad range of mechanisms such as scheduling disciplines, traffic shaping schemes, call admission control and routing algorithms.

This thesis provides a framework for QoS provision, call admission control and routing that is suitable for a wide range of applications and traffic characteristics, as well as for a wide range of scheduling disciplines.

The scheduling disciplines employed in the nodes determine to a large extent the QoS guarantees that can be provided by the network. The basic function of the scheduler is to arbitrate between the packets that are ready for transmission on the link. Based on the algorithm used for scheduling packets and the traffic characteristics of the flows multiplexed on the link, worst-case bounds on the backlog and delay can be computed. These can be used by the network to provide end-to-end QoS guarantees. The corresponding QoS routing problem is, therefore, to find a path which complies with the end-to-end constraints derived from the users’ QoS requirements. Obviously, QoS routing is an essential part of QoS provision. Lack of an efficient QoS routing scheme may create problems such as high call blocking probability, low network utilization and long connection setup delays.

Scheduling disciplines for guaranteed performance service as well as worst-case end-to-end performance bounds have been established and explored in a large number of studies, under both deterministic and stochastic settings (e.g., [1, 2, 3, 4, 5, 6, 7] and references therein). The corresponding routing problems have been addressed as well (e.g., [8, 9, 10, 11]). However, a comprehensive study, which considers both problems, i.e., the establishment of end-to-end performance bounds and the corresponding routing problems, has not been performed. Since these two problems are closely related, moreover the former is a prerequisite for the
latter, in this research they are investigated as a whole. Consider, for example, a routing algorithm which identifies the shortest path with respect to the propagation delays among all paths with the maximal available rate (i.e., the shortest-widest path). Clearly, this routing algorithm is not necessarily adequate for certain scheduling disciplines. With Packetized-GPS networks, for instance, the routing algorithm should account for the number of hops along the path (due to the non-cut-through effect on the end-to-end delay), as well as for the propagation delays and the relation between them and the rate. Furthermore, for certain scheduling disciplines, previous end-to-end performance bounds should be extended to account for packetized traffic and propagation delays. Thus, the goal of this research is to establish (at times, extend) bounds on the end-to-end performance under various settings (i.e., traffic models and scheduling disciplines), and to develop corresponding efficient QoS routing schemes.

This study focuses on the "rate based" class [1, 4] and in particular on the Generalized Processor Sharing - GPS [2] (also known as Fluid Fair Queuing) as well as the Rate-Controlled Earliest Deadline First (RC-EDF) [7] scheduling disciplines. These disciplines provide isolation between sessions so that per session worst-case bounds on the backlog and delay, both for a single server in isolation and for networks with arbitrary topology, can be derived. Given these bounds, the corresponding routing problem is to identify the "best" path with respect to the QoS delay requirements.

First, this study considers the deterministic setting where the source traffic of each session is Burstiness Constrained (BC) (hence, traffic from each session conforms to Cruz’s Linear Bounded Arrival Process (LBAP) traffic model [12]). Under a deterministic setting, the input traffic bursts are assumed to be of bounded length. This is often not the case in practice. Hence, a setting that considers the stochastic nature of the traffic is desired. Under a stochastic setting, only stochastic QoS is guaranteed, i.e., it is guaranteed that the end-to-end delay experienced by a high percentage of the packets does not significantly exceed the required delay. Such guarantees are appropriate for many applications, in particular multimedia applications, which can tolerate a certain amount of loss due to either late arrival or buffer overflow. With stochastic guarantees, tighter bounds and, consequently, better network utilization, can be achieved. Accordingly, in [13], a stochastic approach for bounding the traffic process has been proposed. Rather than assuming that the traffic process has a bounded burstiness, an exponential decay on the distribution of its burst length is imposed. Such a bounded process is called Exponentially Bounded Burstiness (EBB). A more general source traffic model, of Stochastically Bounded Burstiness (SBB) processes, whose burstiness is stochastically bounded by a general decreasing function, was proposed in [14]. These stochastic traffic models, i.e., EBB and SBB, are at the main focus of this study.
Finally, QoS provision and call admission control in fault tolerant networks are considered. Here, the required QoS is *call survivability* in the presence of resource failures. Our study formulates the call admission control problem as a *Markov Decision Process (MDP)* and considers the structure of the optimal call admission policy.

Section 1.1 contains a survey of related work and provides the necessary background. In section 1.2 we overview the specific problems and major results obtained in the course of the research, and explain the structure of the subsequent chapters of the dissertation.

## 1.1 Background and Related Work

QoS provision and routing depend on the scheduling discipline employed in the network. The basic function of the scheduler is to arbitrate between the packets that are ready for transmission on the link. Based on the algorithm used for scheduling packets, as well as the traffic characteristics of the flows multiplexed on the link, certain performance measures can be computed. These can then be used by call admission control and routing algorithms to provide end-to-end QoS guarantees.

First, we describe several common characteristics of the user traffic at the ingress to the network. Next, we provide a brief survey of scheduling disciplines, focusing on the generalized processor sharing and the earliest deadline first disciplines. Next, we deal with the establishment of end-to-end delay bounds. Then, we consider the corresponding routing problems.

### 1.1.1 Traffic Specifications

In traditional queuing theory, most traffic models are based on stochastic processes, e.g., Poisson, on-off, and more sophisticated Markovian processes. In general, these models are either too simple to characterize some important properties of the source or too complex for tractable analysis. An alternative approach is to bound the traffic rather than characterize the process exactly.

Consider, first, the Burstiness Constrained traffic model [12], which is essentially identical to the Leaky Bucket scheme [2]. A traffic stream with rate function $A(t)$, is *Burstiness Constrained (BC)* if, for every $\tau > s > 0$,

$$\int_s^\tau A(t) \, dt \leq \rho (\tau - s) + \sigma,$$

where $\rho$ is the long term upper rate of the arrival process and $\sigma$ is the maximal burst size.
In particular, the TSpec specification proposed for the Internet [15] is based on such constraints. It consists of the following parameters: a token bucket \( p \), a minimum policed unit \( m \), and a maximum datagram size \( M \). The token bucket has a bucket depth, \( b \), and a bucket rate, \( r \). The token bucket, the peak rate and maximum datagram size, together, define the conformance test that identifies the user packets eligible for service guarantees. This test defines the maximum amount of traffic that the user can inject into the network and for which it can expect to receive the service guarantees it has contracted. This maximum amount of traffic is expressed by an upper bound on the amount of traffic generated in any time interval \([s, \tau]\):

\[
\int_s^\tau A(t) \, dt \leq \min\{M + p(\tau - s), b + r(\tau - s)\}.
\]

The TSpec further includes a minimum policed unit \( m \), which counts any packet of size less than \( m \) as being of size \( m \).

Under the Burstiness Constrained traffic model, the bursts of the input process are assumed to be of bounded length. This is not the case for most commonly used input processes (e.g. Bernoulli, Poisson). Thus, in [13], a stochastic approach for bounding the traffic process has been proposed. Rather than assuming that the traffic process has a bounded burstiness, an exponential decay on the distribution of its burst length is imposed. Such a bounded process is called Exponentially Bounded Burstiness (EBB). More precisely, this process, and its generalization, namely the Exponentially Bounded process, are defined as follows. A stochastic process \( A(t) \) is Exponentially Bounded (EB) with parameters \( (\Lambda, \alpha) \), if for any \( t \) and any \( \sigma \geq 0 \), the following bound applies:

\[
\Pr\{A(t) \geq \sigma\} \leq \Lambda e^{-\alpha \sigma}.
\] (1.1)

Let \( A(t) \) be the instantaneous traffic rate. \( A(t) \) has Exponentially Bounded Burstiness (EBB) with parameters \( (\rho, \Lambda, \alpha) \), if for any \( s, \tau \) and any \( \sigma \geq 0 \), the following upper bound, on the tail distribution of the traffic arriving during the time interval \([s, \tau]\), holds:

\[
\Pr\left\{ \int_s^\tau A(t) \, dt \geq \rho(\tau - s) + \sigma \right\} \leq \Lambda \cdot e^{-\alpha \sigma}.
\] (1.2)

A more general source traffic model, of Stochastically Bounded Burstiness processes, whose burstiness is stochastically bounded by a general decreasing function, was proposed in [14]. This approach is based on a generalization of the EBB network calculus, where only exponentially decaying bounding functions were considered. It has two major advantages: (i) it applies to a larger class of input processes and (ii) it provides much better bounds for common models of real-time traffic.
Formally, a stochastic process, \( A(t) \), is *Stochastically Bounded (SB)* with bounding function \( f(\sigma) \) if: (i) \( f(\sigma) \in \Gamma \), and (ii) \( \Pr \{A(t) \geq \sigma\} \leq f(\sigma) \) for all \( \sigma \geq 0 \) and all \( t \geq 0 \), where \( \Gamma \) represents the set of all the functions \( f(\sigma) \) such that, for any order \( n \), the multiple integral \( (\int_{-\sigma}^{\infty} dt)^n f(t) \) is bounded for any \( \sigma \geq 0 \). Accordingly, the rate of a continuous traffic stream \( A(t) \) has *Stochastically Bounded Burstiness (SBB)*, with upper rate \( \rho \) and bounding function \( f(\sigma) \), if: (i) \( f(\sigma) \in \Gamma \), and (ii) \( \Pr \left\{ \int_{s}^{\tau} A(t) dt \geq \rho (\tau - s) + \sigma \right\} \leq f(\sigma) \) for all \( \sigma \geq 0 \) and all \( 0 \leq s \leq \tau \).

The above traffic characteristics are defined for continuous time. Alternatively, a discrete time traffic model can be defined as follows [16]: given a non-decreasing function \( b(\cdot) \), a discrete time sequence \( R \) is said to be \( b \)-smooth if, for all \( m \leq n \) it holds that

\[
R[m+1,n] \leq b(n-m),
\]

where \( R[n] \) denotes the number of packets traversing the link in slot \( n \), and

\[
R[m,n] = \begin{cases} 
\sum_{i=m}^{n} R[i] & \text{if } m \leq n \\
0 & \text{otherwise}
\end{cases}
\]

The function \( b(\cdot) \) is called the *arrival curve*. In the special case where \( b \) is affine, say \( b(x) = \sigma + \rho x \), we say that \( R \) is \( (\sigma, \rho) \)-smooth. The \( (\sigma, \rho) \)-smooth traffic model is the discrete time equivalent of the above BC model.

The definition of smoothness can be generalized to stochastic settings [17]. Accordingly, given a sequence \( \epsilon_0(k) \), \( R \) is said to be \( b \)-smooth with overflow profile \( \epsilon_0 \) if, for all \( k \geq 0 \), we have

\[
\Pr \{W_b(R)[n] > k\} \leq \epsilon_0(k),
\]

where \( W_b(R)[n] \) is the arrival curve conformance process, defined as

\[
W_b(R)[n] = \max_{m:m \leq n} \{R[m+1,n] - b(n-m)\}.
\]

Given a sequence \( \epsilon_e(d) \), \( R \) is said to be \( b \)-smooth with earliness profile \( \epsilon_e \) if, for all non-negative \( d \), it holds that

\[
\Pr \left\{ \max_{m:m \leq n} \{R[m+1,n] - b(n-m+d)\} > 0 \right\} \leq \epsilon_e(d).
\]

### 1.1.2 Scheduling Policies

An important issue in providing performance guarantees is the choice of the scheduling policy and service discipline employed in the switching nodes. In a packet-switching network, packets from different connections interact with each other at
each switch; without proper control, these interactions may adversely affect the network performance experienced by clients. The service disciplines control the order of packet service and determine how the packets from different connections interact.

An overview of several service disciplines that support end-to-end performance guarantees is provided in [1]. We mainly consider the "rate based" class [1, 4, 7], and in particular the Generalized Processor Sharing (GPS) [2, 3] and the Rate-controlled Earliest Deadline First [7] scheduling disciplines. These disciplines provide isolation between sessions so that per session worst-case bounds on the backlog and delay, both for a single server in isolation and for networks with arbitrary topology, can be derived.

A GPS server is work conserving, i.e., it is busy if there are packets waiting in the system. A GPS server is characterized by positive real numbers \( \phi^1, \phi^2, \ldots, \phi^I \), where \( I \) is the number of sessions entering the GPS server. Let \( S^i(\tau, t) \) be the amount of session \( i \) traffic served in an interval \( (\tau, t] \). A GPS server is defined as one with

\[
\frac{S^i(\tau, t)}{S^j(\tau, t)} \geq \frac{\phi^i}{\phi^j}, \quad j = 1, 2, \ldots, I
\]

for any session \( i \) that is continuously backlogged in the interval \( (\tau, t] \). Thus, session \( i \) is guaranteed a service rate of

\[
g^i = \frac{\phi^i}{\sum_{j=1}^{I} \phi^j} R.
\]

A special case of the above is the Rate Proportional Processor Sharing (RPPS) GPS assignment, where \( \phi^i = \rho^i \), 1 \( i \leq I \), and \( \rho^i \) is the session upper rate.

For the GPS service discipline, it is assumed that the server can serve multiple sessions simultaneously and that the traffic is infinitely divisible. A PGPS (originally named Weighted Fair Queueing) server approximates the GPS service in a packetized system. With PGPS, when the server is ready to transmit the next packet at time \( \tau \), it picks, among all the packets queued in the system at time \( \tau \), the first packet that would complete service in the corresponding GPS system if no additional packets were to arrive after time \( \tau \). Let \( S^i(\tau, t) \) and \( \hat{S}^i(\tau, t) \) be the amount of session \( i \) traffic served under GPS and PGPS (correspondingly) in the interval \( (\tau, t] \). Then, for all times \( \tau \) and session \( i \) [2]:

\[
S^i(0, \tau) - \hat{S}^i(0, \tau) \leq L_{\text{max}}.
\]

The need to emulate a reference GPS server in a PGPS server is computationally expensive. One simpler packet approximation algorithm of GPS is Self-Clocked Fair Queueing (SCFQ) [18]. In [19], a Start-Time Fair Queueing (STFQ) algorithm that
is computationally efficient and achieves fairness regardless of variation in a server capacity is proposed. It is shown there that PGPS may become unfair over variable rate servers, whereas the proposed start-time fair queueing algorithm retains fairness both for Fluctuation Constrained (FC) and Exponentially Bounded Fluctuation (EBF) servers [20].

With work-conserving disciplines, the traffic pattern is distorted inside the network due to network load fluctuation, and there are a number of difficulties and limitations in deriving the traffic characterization after the distortion. Alternatively, we can control the distortions at each switch using non-work-conserving disciplines. With such a discipline, the server may be idle even when there are packets waiting to be sent. An important discipline among the class of non-work-conserving disciplines is the Rate-Controlled Earliest-Deadline-First (EDF) [7].

An EDF scheduler assigns each arriving packet a time-stamp corresponding to its deadline. The EDF scheduler maintains a single queue of untransmitted packets, which are sorted by increasing order of deadlines. The scheduler always selects the packet in the first position of the queue, that is, the packet with the lowest deadline, for transmission. In the case of a single node, EDF is known to be an optimal scheduling policy in terms of schedulable region for a set of flows with given traffic envelopes and deterministic delay requirements [4].

1.1.3 End-to-End Guarantees

The provision of QoS guarantees, such as delay, depends on the establishment of relatively tight and simple end-to-end bounds. Loose bounds may result in low network utilization, whereas complex bounds cannot be easily used as part of call admission control and routing algorithms. Such end-to-end bounds are mainly determined by (i) the characterization of the connections traffic, and (ii) the packet scheduling disciplines at each server or switch in the network.

Delay analysis techniques can be grouped into two classes, depending on whether they decompose the network into isolated servers that are analyzed separately (the decomposition approach [6, 12, 21]), or whether they integrate individual servers in the network into larger super-servers (the service-curve approach [2, 3, 16]).

Worst-case bounds on the end-to-end delay for BC traffic in arbitrary topology networks of GPS servers was first introduced in [3]. Through a service-curve based analysis, the following closed-form end-to-end delay bound was derived there for a BC traffic in networks of RPPS PGPS servers:

\[
D^i \leq \frac{\sigma^i + n(p)L^i}{r^i(p)} + \sum_{l \in p} \left( \frac{L_{\text{max}}}{R_l} + d_l \right),
\]

where \(\sigma^i\) is session \(i\)'s maximal burst size, \(L^i\) is its maximal packet size, \(L_{\text{max}}\) is the
maximal packet size in the network, \( r^i(p) \) is the guaranteed service rate for session \( i \) along the path \( p \), \( n(p) \) is the number of hops (servers) along \( p \), \( R_l \) is the service rate of a link \( l \), and \( d_l \) is the propagation delay of \( l \).

A Consistent Relative Session Treatment (CRST) GPS assignment is an assignment for which there exists a strict ordering of the sessions such that, for any two sessions \( i, j \), if session \( i \) is less than session \( j \) in the ordering, then session \( i \) does not impede session \( j \) at any node of the network. Here, session \( j \) is said to impede session \( i \) at a link \( l \) if \( \frac{\phi^i_l}{\phi^j_l} < \frac{\rho^i}{\rho^j} \).

Closed-Form deterministic end-to-end delay bounds for the broader class of CRST GPS networks were established in [22]. These bounds were obtained through a decomposition-based delay analysis. This analysis is quite general, and can be employed also in the study of GPS networks in a stochastic setting. Indeed, the decomposition based analysis was employed in [6] to derive upper bounds on the tail distribution of session backlog and delay, both for a single GPS server in isolation, as well as for RPPS GPS networks. These bounds were established using Exponentially Bounded Burstiness (EBB) processes as source session traffic models.

Assume that packets are infinitely divisible (fluid model) and that propagation delays are negligible, i.e., the end-to-end delay is solely the outcome of the delays in the queues and processing time. It was shown in [6] that, if every session \( i \) in a RPPS GPS network is an EBB process with parameters \((\rho^i, \Lambda^i, \alpha^i)\), then, at any time \( t \) and for any \( D > 0 \),

\[
\Pr \{ D^i_p(t) \geq D \} \leq \Lambda^i_p e^{-\alpha^i r^i(p) D},
\]

where

\[
\Lambda^i_p = \frac{\Lambda^i e^{\alpha^i \rho^i \xi}}{1 - e^{-\alpha^i (r^i(p) - \rho^i) \xi}}, \quad 0 < \xi < \frac{\ln (\Lambda^i + 1)}{\alpha^i (r^i(p) - \rho^i)},
\]

and \( r^i(p) \) is the guaranteed service rate for session \( i \) along the path \( p \).

A stochastic generalization of the service curve concept was presented in [17]. There, stochastic bounds on delay and backlog of a network element in isolation were derived for a \( b \)-smooth with earliness profile \( \epsilon_e \) input traffic. A network analysis (the multiple node case) was not provided, neither were bounds for the simpler EBB traffic model considered.

Consider now non-work-conserving service disciplines. Since a packet may be held in the server even when the server is idle, average delay of packets may increase and the average throughput of the server may decrease. However, our main concern is the end-to-end delay bound rather than these average values. Furthermore, non-work-conserving service disciplines greatly simplify the analysis in a network
environment by allowing a single node analysis to be extended to arbitrary topology networks. Thus, non-work conserving service disciplines are very attractive candidates for providing end-to-end performance guarantees. Non-work-conserving schedulers, combined with rate-controllers, can be expressed by a general class of disciplines called rate-controlled service disciplines [7]. In this class of service disciplines, the traffic of each connection is reshaped at every node to ensure that the traffic offered to the scheduler conforms with specific characteristics. In particular, typical regulators enforce, at each server inside the network, the same traffic parameter control as the one performed at the network access point. Reshaping makes the traffic at each node more predictable and, therefore, simplifies the task of guaranteeing performance to individual connections. When used with a particular scheduling policy, end-to-end delay bounds can be computed as the sum of the worst-case delay bounds at each server along the path [7]. The main advantages of a rate-controlled service discipline, especially when compared to GPS, are simplicity of implementation and flexibility. In [4], it was shown that any end-to-end delay bounds that can be guaranteed by the GPS discipline, can also be achieved by a rate-controlled discipline, by using a simple algorithm to determine how to reshape the traffic, and then specify worst-case delay bounds at each server. The sum of the worst-case delay bounds of this rate-controlled discipline is no larger than the end-to-end bound provided by the GPS discipline. In particular, it was shown in [4] that the use of shaper parameters induced by GPS allows Rate-Controlled EDF (RC-EDF), i.e., the EDF scheduling policy with per-hop reshaping, to outperform GPS. The design of shapers that achieve even larger schedulable regions has been addressed in [23].

1.1.4 Routing Algorithms

With end-to-end delay bounds at hand, efficient QoS routing algorithms that exploit these bounds are called for. Here, by "efficient" we mean that the algorithm is of low complexity and is either optimal (in some sense, e.g. consumes minimum network resources) or near-optimal. Indeed, QoS routing has been the subject of several studies and proposals [8, 9, 10, 24, 25].

Unlike traditional routing algorithms that usually account for a single metric, such as hop-count or delay, QoS routing must consider multiple metrics such as cost, bandwidth, and delay. However, finding a path subject to multiple constraints is inherently (i.e., NP-) hard, hence, in general, optimal solutions of polynomial complexity should not be sought. Even the simple problem of determining whether there exists a path whose cost and delay are within some given bounds, i.e., the Restricted Shortest Path problem (RSP), is NP-complete [26]. In [24] it was proven
that the problem of deciding if there is a simple path for which $n \geq 2$ additive metrics (e.g. delay and cost) or multiplicative metrics (e.g. the probability of successful transmission) are within given constraints is NP-complete. It was also shown that the similar problem with $n \geq 1$ additive metrics and $k \geq 1$ multiplicative metrics is NP-complete as well.

The QoS routing problem, of identifying the best path with respect to the QoS delay requirements, in networks with rate-based schedulers, has been the subject of several studies, e.g. [8, 9, 10, 25]. In particular, it was shown in [8] that, for a given connection and end-to-end delay constraint, the existence and identity of a feasible path can be obtained through up to $M$ executions of a standard shortest path algorithm, where $M$ is the number of network links (a similar result was reported in [10]). In [9], a rate quantization method was employed, to establish a near-optimal solution for the basic problem of identifying a feasible route. The more general problem, of optimizing the route choice in terms of balancing the loads and accommodating multiple connections was considered as well, and several schemes were proposed. It was also suggested (but not established) that rate quantization can be applied to these schemes too, in order to reduce their complexity.

Call admission control and routing algorithms in networks that employ non-work-conserving service disciplines require schedulability conditions that detect violations of delay guarantees in a network switch. Exact schedulability conditions for EDF disciplines have been presented in [27]. Call admission control algorithms for the RC-EDF service discipline have been studied in [28]. Although the EDF schedulability condition in [27] can be expressed simply, the algorithm to perform these schedulability tests can be computationally complex, or, in the general case, require an unbounded number of values that must be checked. In [28], simple and computationally efficient algorithms for performing flow admission at links have been presented. Channel establishment algorithms (i.e. route selection and resource reservation along the route), for networks that employ RC-EDF schedulers have been considered in [11, 29, 30].

In [29], a single-round-trip procedure for establishing channels has been devised. Accordingly, when a node receives an establishment request message, it performs tests that are concerned with the availability of sufficient bandwidth in the links and with the compliance with the end-to-end delay guarantees. If any test fails at a node, the channel cannot be established along that route; the message is sent back, either to the sender or to an intermediate node that can try sending the message towards the destination along another path. If all tests succeed at all nodes and at the destination host, this host subdivides the delay bound among the nodes traversed by the channel, after subtracting the link delays along the route. Then, the destination host sends a reply message back to the source node along the channel’s
route.

In [11], a table-driven distributed route-selection scheme that is guaranteed to find a “qualified” route has been proposed. “Qualified” refers to a route that meets the performance requirements of the requested channel without compromising any of the existing guarantees. Accordingly, the Bellman-Ford algorithm is applied to build, on each node, a loop-free table, based on a Minimum Worst-case Response Time (MWRT) [30]. When the source node wishes to establish a real-time channel, it will try to find the current least-MWRT route. Since the sum of MWRTs over all links on the path from source to destination may be smaller than the requested end-to-end delay, it is allowed to spend more time than the corresponding MWRT when sending a message over each intermediate link. In [11] it is proposed to divide the “extra” delay along the route, either evenly or in proportion to each link’s MWRT. The authors of [31] have proposed a number of resource division policies used for mapping the end-to-end delay requirement of a call into local delay deadlines to be reserved at each scheduler.

1.2 Overview of Major Results

The results of our study are presented in Chapters 2-5. For convenience, the chapters are arranged as independent units, so that they can be read in any order. In particular, each chapter has its own introduction and conclusion sections, describing the related work and the importance of the results with the specific context of that chapter. To maintain presentation continuity, some of the proofs are omitted from the main text and gathered as an appendix at the end of the relevant chapter.

In Chapter 2, we focus on the GPS scheduling discipline. First, we study the case of burstiness-constrained (BC) traffic with deterministic QoS requirements. Here, we extend the results of a previous study in order to obtain routing schemes of low computational complexity that identify feasible paths while optimizing some network utilization criteria. Next, we consider traffic with exponentially bounded burstiness (EBB) and stochastic QoS requirements. Here, we extend previous results and provide a bound on the distribution of the end-to-end delay for packetized traffic and links with non-negligible propagation delays. Consequently, we formulate several routing schemes that identify feasible paths under various network optimization criteria. Then, we consider traffic with (general) stochastic bounded burstiness (SBB). Here, we provide the corresponding end-to-end bound for packetized traffic and links with propagation delays. Then, focusing on the special case of a bounding function that is the sum of exponents, we design appropriate routing schemes. Finally, we investigate variable-rate links. In both the deterministic as well as the stochastic settings, we extend previous results, obtained for a single-input FCFS
server in isolation, and establish end-to-end bounds for a complete network, packetized traffic and non-negligible propagation delays. With these bounds at hand, we formulate appropriate routing schemes.

In Chapters 3 and 4, we turn our attention to the rate-controlled EDF scheduling discipline, for which traffic is reshaped at each node along the path. In Chapter 3, assuming BC traffic with deterministic QoS requirements, we study the joint problem of identifying a feasible path and optimizing the reshaping parameters along the path. Here, we broaden the space of feasible solutions by allowing to reshape traffic with different parameters at each hop. Then, we turn to consider the problem of optimizing the route choice in terms of balancing the loads and accommodating multiple connections. Next, in Chapter 4, assuming the EBB traffic model, we establish results that extend the deterministic study of rate-controlled EDF, both for a single server in isolation and for networks of servers. First, we establish schedulability conditions under the stochastic setting and show the optimality of the EDF discipline in terms of schedulable regions. In the general (multi-hop) setting, we first establish upper bounds on the tail distribution of the end-to-end delay for traffic shaper elements in series and then establish upper bounds for rate-controlled EDF networks. Finally, we show that the guaranteed upper bound on the end-to-end delay tail distribution in GPS networks can be guaranteed in rate-controlled EDF networks as well.

In Chapter 5, we consider the provision of QoS in fault tolerant networks. Here, the required QoS is given in terms of call termination probability upon resource failure. In this study, we consider the optimal admission control policies for two network architectures, namely, optical networks with shared protection and wireless networks.

Finally, in Chapter 6, we conclude our study and outline possible directions for future research.
Chapter 2

The GPS scheduling discipline

2.1 Introduction

The ability to support QoS requirements depends on the scheduling policies employed in the nodes. The Generalized Processor Sharing - GPS [2, 3] scheduling discipline, which is considered in this chapter, provides isolation between sessions so that per session worst-case bounds on the backlog and delay, both for a single server in isolation and for networks with arbitrary topology, can be derived. Given these bounds, the corresponding routing problem is to identify the "best" path with respect to the QoS delay requirements.

Under a deterministic setting, QoS provision and routing have been the subject of several studies, as described in section 1.1. In particular, in [9], a rate quantization method was employed, to establish a near-optimal solution for the basic problem of identifying a feasible route. The more general problem, of optimizing the route choice in terms of balancing the loads and accommodating multiple connections was considered as well, and several schemes were proposed. It was also suggested (but not established) that rate quantization can be applied to these schemes too, in order to reduce their complexity. Such rate quantization schemes, under a deterministic setting, are the subject of the first part of this chapter.

As described in section 1.1, the statistical behavior of the GPS scheduling discipline using Exponentially Bounded Burstiness (EBB) processes [13] as source session traffic models, was studied in [6], and upper bounds on the tail distribution of session backlog and delay were derived, both for a single GPS server in isolation, as well as for Rate Proportional Processor Sharing (RPPS) GPS networks. [6] focused solely on “fluid” (non-packetized) GPS networks. We extend those results to packetized traffic and incorporate deterministic propagation delays in the network model. With these extended upper bounds at hand, we study the corresponding QoS routing problem. It turns out that this problem, though more complicated, resembles the routing problem under the deterministic setting.
In order to deal with processes that do not comply with the EBB characterization, a more general framework, of Stochastically Bounded Burstiness (SBB) traffic, was developed in [14], for an isolated network element. The SBB calculus is also a powerful tool for obtaining much tighter bounds for multiple time-scale processes. We adopt the SBB calculus of [14], and extend it in order to obtain bounds on the end-to-end delay tail distribution for a (packetized) RPPS PGPS network with propagation delays. Due to the complexity of the bounds, we establish routing schemes only for a special case of the SBB model, in which the bounding function is the sum of two exponents. The advantage of this special case over the EBB calculus is demonstrated through a simulation example of a simple network.

Finally, we consider networks with variable-rate links. In [20], a single-input variable-rate server in isolation was investigated. We extend those results in order to obtain deterministic and stochastic end-to-end delay bounds for a packetized traffic entering a network with variable-rate links and propagation delays. Then, we establish corresponding routing algorithms under both the deterministic as well as the stochastic settings.

We note that our network analysis is based on the decomposition approach [6] and not on the statistical service curve approach [17, 32]. The statistical service curve approach [17, 32] provides a reasonably general framework for the statistical analysis of a variety of scheduling schemes, including the GPS discipline. It is based on a stochastic extension of the “service curve” concept developed by Cruz [12]. However, as observed in [33], it is not clear whether this framework can be extended to the multi-node case.

Our stochastic model differs from that of [32, 33, 34, 35]. In our case, the stochasticity of the end-to-end guarantees is due to the stochastic nature of the session input traffic, whereas in [32, 33, 34, 35], the exploitation of statistical multiplexing results in a provision of (only) stochastic guarantees even for deterministically bounded input traffic.

The rest of the chapter is structured as follows. In Section 2.2, we formulate the model. Next, in Section 2.3, we consider routing schemes under a deterministic setting and illustrate the application of the rate quantization method, which reduces the complexity at the expense of performance. Furthermore, we consider networks with variable-rate links and deterministic guarantees. In Section 2.4, we turn to consider the stochastic setting, and in particular, EBB processes. Here, we establish bounds on the end-to-end delay tail distribution for a PGPS network with non-negligible propagation delays; then, we propose corresponding routing schemes and illustrate their efficiency through simulation examples. Next, we consider the more general stochastic framework of SBB processes: we establish bounds on the end-to-end delay tail distribution and propose a routing scheme for a special case. Then,
we consider networks with variable-rate links and stochastic guarantees. Finally, in Section 2.5, we conclude the chapter. To maintain presentation continuity, some of the proofs are omitted from the main text and gathered as an appendix in Section 2.6.

### 2.2 Model Formulation

Given is a network across which sessions need to be routed. The network is represented by a directed Graph $G(V, E)$, in which nodes represent switches and arcs represent links. $V$ is the set of nodes and $E$ is the set of interconnecting links; let $|V| = N$ and $|E| = M$.

Each link $l \in E$ is characterized by (i) a service rate $R_l$ and (ii) a constant delay value $d_l$, related to the link’s speed, propagation delay and maximal transfer unit. We assume that the Packetized Generalized Processor Sharing (PGPS) scheduling discipline is employed in each link $l \in E$ to ensure a guaranteed share of link resources. Since each link is associated with a (PGPS) server, we shall henceforth use the terms “server” and “link” interchangeably. We note that, if the service rate is a nodal property, then we associate it with all its outgoing links.

As done in, e.g., [8, 9, 10, 36, 37], we map end-to-end delay guarantees into rate assignments. Namely, we assign the GPS parameters according to the required end-to-end delay. We assume that, for each session $i$, the GPS assignment $\phi^i_l$ is proportional to the required rate $r^i$, rather than to the actual session traffic upper rate $\rho^i$, i.e., $\phi^i_l = r^i > \rho^i$ and $\sum_{j=1}^{l} r^j < R_l$. Thus, each session $i$ is guaranteed a service rate of $g^i_l = \frac{r^i}{\sum_{j=1}^{l} r^j} \cdot R_l > r^i$. Obviously, with such an assignment, all sessions are locally stable [2] and we can follow the same network analysis as that of an RPPS-GPS network.

Denote by $R'_l$ the maximal available rate which link $l$ can offer to a new connection. More formally, $R'_l = R_l - \sum_{j=1}^{l} r^j$, where $R_l$ is the link’s service rate. When a new connection with a rate $r < R'_l$ is established through link $l$, the value of $R'_l$ becomes $R'_l - r$. Let $R^1 \leq R^2 \leq \ldots \leq R^K$ be the different values of $R'_l$, for all $l \in E$; clearly, $K \leq M$.

Following [8, 10, 9], we assume a source (“explicit”) QoS routing framework, in which link state information is exchanged and maintained up-to-date among network nodes for path computation. Routing decisions are based on the image of the network at the source node.

A session $i$ in the network is characterized by the following parameters:

- Source and destination nodes.
• Some stochastic bounds on the session’s burst.
• A traffic upper rate $\rho^i$, which constitutes the minimal bandwidth requirement.
• A maximum packet size $L^i$.
• A required end-to-end delay $D^i$.
• A “sensitivity” to end-to-end delay fluctuations $q^i$.

A session should be routed through some path $p$ between the corresponding source and destination nodes. We denote by $n(p)$ the number of hops (i.e., links) of a path $p$, and by $r(p)$ its maximal available rate, i.e., $r(p) = \min_{l \in p} R^i_l$. When a session $i$ is routed over a path $p$ with a reserved rate $r$ (where $r \leq r(p)$), the end-to-end delay, $D^i(p, r)$, is stochastically bounded. We also denote by $D^i(p)$ the minimal value of $D^i(p, r)$, i.e., $D^i(p) = D^i(p, r(p))$.

Let $g^i_p$ be the guaranteed service rate for session $i$ along $p$, i.e. $g^i_p = \min_{l \in p} g^i_l$, where $g^i_l$ is the guaranteed service rate at a server $l \in p$. Let $I(l)$ denote the set of sessions present at server $l$. For every session $i \in I(l)$, denote the arrival process into server $l$ by $A^i_l$ and the departure process by $S^i_l$. Let $A^i_l$ be the arrival process of session $i$ at the first hop on its path, and $A^i_{l(n)}$ be the arrival process at the $n$-th hop. Thus, when a link $l$ is the $n$-th hop on a path $p$ of a particular session $i$, the processes $A^i_l$ and $A^i_{l(n)}$ are identical. The departure process of a session $i$ at the $n$-th hop is denoted by $S^i_{l(n)}$, while $S^i_{out} = S^i_{l(n(p))}$ describes its departure process from the network. The backlog of session $i$ at the $n$-th hop at time $t$ is denoted by $Q^i_{l(n)}(t)$, and the total backlog over the path $p$ at time $t$ is denoted by $Q^i_p(t)$. Finally, the instantaneous end-to-end delay of session $i$ at time $t$ is denoted by $D^i_p(t)$.

We assume that each session is associated with a certain probability $q^i$, which reflects its “sensitivity” to end-to-end delay fluctuations beyond the required delay. Accordingly, a path is said to be $q$-feasible if the end-to-end delay fluctuations beyond the required value conforms with the session’s sensitivity.

**Definition 2.1** The effective end-to-end delay $\bar{D}^i(p, q^i)$ of a path $p$, is the maximal end-to-end delay for which the tail distribution is guaranteed to be at most $q^i$, i.e., $\Pr \{ D^i(p) \geq \bar{D}^i(p, q^i) \} \leq q^i$.

**Definition 2.2** A path $p$ is $q$-feasible for session $i$ if the end-to-end delay tail distribution beyond the required delay $D^i$ is at most $q$, i.e., $\Pr \{ D^i(p) \geq D^i \} \leq q$. Alternatively, a path $p$ is $q$-feasible if the effective delay $\bar{D}^i(p, q^i)$ of the path $p$ is at most the required delay $D^i$, i.e., $\bar{D}^i(p, q^i) \leq D^i$. 20
A session $i$ is $q$-feasible if it has a $q$-feasible path. Finally, a path with the minimal effective delay $\bar{D}(p,q)$ is termed quickest.

### 2.3 Deterministic Setting

In this section we consider QoS routing under a deterministic setting and extend the results presented in [9]. We note that the main contribution of this chapter is reported in the next sections; the main purpose of the present section is to provide some foundations which facilitate the presentation in the next sections.

In [9], optimal solutions were proposed for the basic problem of identifying a feasible path and for more general problems of optimizing the route choice in terms of balancing the loads. For the basic problem, a rate quantization method was applied, in order to reduce complexity at the expense of performance. The rate quantization approach was not investigated (albeit suggested) for the optimization problems. Accordingly, we apply this approach to two schemes, the first aims at reducing the consumption of rate, and the second aims at balancing the loads. It turns out that these schemes are more complex than the rate-quantized solution for the basic routing problem.

In this section we assume that the input traffic is Burstiness Constrained [12]. Accordingly, when a connection is routed over a path $p$ with a guaranteed rate $r$ (where $r \leq r(p)$), the following upper bound $D(p,r)$ on the end-to-end delay applies:

$$D(p,r) = \frac{\sigma + n(p)L}{r} + \sum_{l \in p} d_l,$$

where $n(p)$ is the number of hops of a path $p$, $L$ is the maximal packet size and $d_l$ is the propagation delay at link $l$.

A path $p$ is said to be feasible if there exists a value $r, \rho \leq r \leq r(p)$, such that $D(p,r) \leq D$, or alternatively, if $D(p,r(p)) \leq D$ and $r(p) \geq \rho$. A connection $i$ is feasible if it has a feasible path.

Recall that the available rate values in the network are denoted by $R^1, R^2, \ldots, R^K$, where $K \leq M$, and let $\hat{R} = \frac{R^K}{M}$. As in [9], our aim is to establish efficient $\epsilon$-optimal routing schemes. Accordingly, given a value $\epsilon > 0$, the available rate values are grouped into $O\left(\log_{1+\epsilon}(\hat{R})\right)$ rate classes, such that for $0 \leq j \leq \lfloor \log_{1+\epsilon}(\hat{R}) \rfloor$, rate class $j$ contains all maximal rates in the range $R^1 (1 + \epsilon)^j \ldots R^1 (1 + \epsilon)^{j+1}$. We say that a link $l$ is in rate-class $j$ if the value of $r_l$ is in that class rate.
2.3.1 Rate Consumption Criterion

In order to accommodate multiple connections throughout the network, the routing algorithm should economize the consumption of rates. Accordingly, consider the following problem.

**Minimum Rate Problem:** Given are: (i) a network $G(V, E)$, with available rates $r^l \in \{R^1, R^2, \ldots, R^K\}$, and a constant delay $d_l$ for each $l \in E$; and (ii) a connection with source $s$, destination $t$, long term upper rate $\rho$, burst $\sigma$, maximal packet size $L$ and delay constraint $D$. Find a feasible path $p$ for which the consumed rate is minimal.

A solution to this problem, with complexity of $O(M \cdot H \cdot K)$, was proposed in [9]. In Figure 2.1, we formulate a rate-quantized solution that finds a near-optimal path with complexity of $O(M \cdot H \cdot \min\{1/\epsilon \cdot \log \hat{R}, K\})$, where $\hat{R} = \frac{R^K}{R^1}$.

---

1. For $j \leftarrow 0$ to $\lfloor \log_{1+\epsilon} \hat{R} \rfloor$
   
   (a) if rate-class $j$ is empty then skip to the next value of $j$
   
   (b) delete from the network all links whose rate-class is less than $j$
   
   (c) for all $n$, $0 \leq n \leq H$ : Find a path $p(n, j)$ that is shortest with respect to the metric $\{d_l\}$, among $n$-hops paths
   
   (d) compute $D(p(n, j)) = \frac{\sigma + n \cdot L}{(1+\epsilon)^j \cdot R^1} + \sum_{l \in p(n, j)} d_l$
   
   (e) if $D(p(n, j)) \leq D$ then
      
      $$r_{\text{min}}(p(n, j)) \leftarrow \frac{\sigma + n \cdot L}{D - \sum_{l \in p(n, j)} d_l}$$
   
   (f) else
      
      i. if $D(p(n, j)) \leq (1 + \epsilon) \cdot D$ then
         
         $$r_{\text{min}}(p(n, j)) \leftarrow (1 + \epsilon)^j \cdot R^1$$
      
      ii. else $r_{\text{min}}(p(n, j)) \leftarrow \infty$

2. let $\tilde{n}$ and $\tilde{j}$ be such that $r_{\text{min}}(p(\tilde{n}, \tilde{j}))$ is minimal over all $1 \leq n \leq H$, $0 \leq j \leq \lfloor \log_{1+\epsilon} \hat{R} \rfloor$. If $r_{\text{min}}(p(\tilde{n}, \tilde{j})) = \infty$ then the connection is not feasible, otherwise $\tilde{p} = p(\tilde{n}, \tilde{j})$ and $r_{\text{min}}(\tilde{p})$ are the required $\epsilon$-optimal path and rate, correspondingly

Figure 2.1: Algorithm Minimum Rate - Rate Quantized (MR-RQ)
Proposition 2.1

1. Suppose that a feasible path exists, and let \( p^* \) and \( r_{\text{min}}(p^*) \) be the optimal path and minimal rate, respectively. Then, the path \( \tilde{p} \) identified by Algorithm MR-RQ, constitutes an \( \epsilon \)-optimal solution, i.e.:

\[
D(\tilde{p}) \leq (1 + \epsilon) \cdot D,
\]

and

\[
\frac{r_{\text{min}}(p^*)}{r_{\text{min}}(\tilde{p})} \geq \frac{D(\tilde{p})}{D}.
\]

2. Suppose that, among the set of \( O\left(\frac{1}{\epsilon} \cdot \log \tilde{R} \cdot H\right) \) paths identified by the algorithm, there is a path \( \bar{p} \) such that:

\[
(1 + \epsilon)^j \cdot R^1 < r_{\text{min}}(\tilde{p}) < (1 + \epsilon)^{j+1} R^1,
\]

(2.1)

then, \( \bar{p} \) is a feasible path, i.e. \( D(\bar{p}) \leq D \), and

\[
r_{\text{min}}(\bar{p}) \leq r_{\text{min}}(\tilde{p}) < (1 + \epsilon) \cdot r_{\text{min}}(p^*) .
\]

3. If \( r_{\text{min}}(p) = \infty \) then the connection is not feasible.

The first part of Proposition 2.1 implies that, when a feasible path exists, the algorithm finds an \( \epsilon \)-feasible path, i.e., \( D(\tilde{p}) \leq (1 + \epsilon) \cdot D \), the second part implies that, occasionally, when inequality (2.1) holds, the algorithm identifies a feasible path \( \bar{p} \), which is \( \epsilon \)-optimal with respect to the consumed rate, i.e.,

\[
r_{\text{min}}(\bar{p}) < (1 + \epsilon) \cdot r_{\text{min}}(p^*) .
\]

Proof.

1. Assume \( p^* \) is the optimal path and let \( j^* \) be the rate-class of the minimal rate value, \( r_{\text{min}}(p^*) \), i.e., \((1 + \epsilon)^{j^*} \cdot R^1 \leq r_{\text{min}}(p^*) < (1 + \epsilon)^{j^*+1} R^1 \). Let \( n^* = n(p^*) \). Consider the path \( p(n^*, j^*) \), identified at the \( j^* \)-th iteration of the algorithm for the value of \( n^* \) hops, we have:

\[
(1 + \epsilon) \cdot D = (1 + \epsilon) \cdot \left( \frac{\sigma + n^* \cdot L}{r_{\text{min}}(p^*)} + \sum_{l \in p^*} d_l \right) \geq \frac{\sigma + n^* \cdot L}{(1 + \epsilon)^{j^*} \cdot R^1} + \sum_{l \in p(n^*, j^*)} d_l.
\]

(2.2)
Thus, condition (1(f)i) in the algorithm holds, and

\[ r_{\min}(p(n^*, j^*)) = (1 + \epsilon)^{j^*} \cdot R^1. \]

Clearly, by the algorithm,

\[ r_{\min}(\hat{p}) \leq (1 + \epsilon)^{j^*} \cdot R^1 \leq r_{\min}(p^*). \]

Consider the two options:

(a) \( \hat{p} = p(n^*, j^*), \) thus,

\[ r_{\min}(\hat{p}) = (1 + \epsilon)^{j^*} R^1 \leq r_{\min}(p^*), \]

(b) \( \hat{p} \neq p(n^*, j^*), \) thus,

\[ r_{\min}(\hat{p}) \leq \frac{1}{(1 + \epsilon)} \cdot (1 + \epsilon)^{j^*} R^1 \leq \frac{1}{(1 + \epsilon)} \cdot r_{\min}(p^*). \]

If \( \hat{p} = p(n^*, j^*) \) then

\[
\frac{r_{\min}(\hat{p})}{r_{\min}(p^*)} \cdot \frac{D(\hat{p})}{D} = \frac{(1 + \epsilon)^{j^*} R^1}{r_{\min}(p^*)} \cdot \frac{D(p(n^*, j^*))}{D} = \frac{\sigma + n^* \cdot L}{(1 + \epsilon)^{j^*} R^1} \cdot \frac{1}{r_{\min}(p^*)} + \sum_{l \in p(n^*, j^*)} d_l \leq \frac{\sigma + n^* \cdot L}{(1 + \epsilon)^{j^*} R^1} + \sum_{l \in p^*} d_l \leq 1. \] (2.3)

If \( \hat{p} \neq p(n^*, j^*) \) then,

\[
\frac{r_{\min}(\hat{p})}{r_{\min}(p^*)} \cdot \frac{D(\hat{p})}{D} \leq \frac{1}{(1 + \epsilon)} \cdot \frac{r_{\min}(p^*)}{r_{\min}(p^*)} \cdot \frac{D(\hat{p})}{D} \leq 1. \]

2. The inequality

\[ (1 + \epsilon)^{j^*} \cdot R^1 < r_{\min}(\hat{p}) < (1 + \epsilon)^{j^*+1} R^1, \]

implies that condition (1e) in the algorithm holds, therefore

\[ D(\hat{p}) \leq D \]

and

\[
r_{\min}(\hat{p}) = (1 + \epsilon)^{j^*} R^1 \leq r_{\min}(p^*) \leq r_{\min}(\hat{p}) < (1 + \epsilon)^{j^*+1} R^1 \leq (1 + \epsilon) \cdot r_{\min}(p^*) \] (2.4)
3. Suppose that a feasible path \( p \) exists. It is clear that the algorithm identifies a path \( \tilde{p} \) with \( D(\tilde{p}) \leq (1 + \epsilon) \cdot D \), which contradicts the fact that \( r_{\min}(p(\tilde{n}, \tilde{j})) = \infty \).

The above MR-RQ algorithm is based on the iterative execution of a Bellman-Ford shortest path procedure. It can be easily shown that a similar algorithm, but which is based on Dijkstra’s shortest path procedure, finds a near optimal path with complexity of

\[
O\left((N \log N + M) \cdot \frac{1}{\epsilon} \cdot \log \hat{R}\right).
\]

The later is of lower complexity when \( (N \log N + M) < M \cdot H \), however it may be inferior when \( \frac{1}{\epsilon} \cdot \log \hat{R} > K \).

2.3.2 Load Balancing Criterion

A better measure for balancing loads over the network may be the relative (rather than absolute) rate consumption [9]. The problem can be stated as finding a feasible path \( p \) that minimizes the value of \( \max_{l \in p} r_{\min} r_l \). The algorithm Minimum Relative Rate (MRR), introduced in [9], solves this problem with complexity of \( O(K \cdot H \cdot M) \). We employ the rate-quantization method to achieve a near-optimal solution of lower complexity to this problem, as well. The complexity of Algorithm Minimum Relative Rate - Rate Quantized, depicted in Figure 2.2, is \( O\left(M \cdot H \cdot \min\left\{\frac{1}{\epsilon} \cdot \log \hat{R}, K\right\}\right) \).

1. for \( j \leftarrow 0 \) to \( \lfloor \log_{1+\epsilon} \hat{R} \rfloor \)
   (a) if rate-class \( j \) is empty then skip to the next value of \( j \)
   (b) delete from the network all links whose rate-class is less than \( j \)
   (c) for all \( n, 0 \leq n \leq H \) : Find a path \( p(n,j) \) that is shortest with respect to the metric \( \{d_l\} \), among \( n \)-hops paths
   (d) \( MRR(n,j) \leftarrow \frac{\sigma + n \cdot \sum_{l \in p(n,j)} d_l}{(1+\epsilon)^{\cdot} \cdot \hat{R} \cdot D \sum_{l \in p(n,j)} d_l} \)

2. let \( \tilde{n} \) and \( \tilde{j} \) be such that \( MRR(\tilde{n}, \tilde{j}) \) is minimal over all \( 1 \leq n \leq H \) and \( 0 \leq j \leq \lfloor \log_{1+\epsilon} \hat{R} \rfloor \)

Figure 2.2: Algorithm Minimum Relative Rate - Rate Quantized (MRR-RQ)
Proposition 2.2 Let \( \hat{p} \), \( \hat{MRR} \) be the output path and relative rate, respectively, identified by Algorithm MRR-RQ, and let \( p^* \) and \( MRR^* \) be the optimal path and minimal relative rate.

1. \( \hat{MRR} < (1 + \epsilon) \cdot MRR^* \).

2. If \( \hat{MRR} \leq 1 \) then the path \( \hat{p} \) identified by the algorithm is feasible.
   If \( 1 < \hat{MRR} < (1 + \epsilon) \) then \( D(\hat{p}) \leq (1 + \epsilon) \cdot D \),
   else if \( \hat{MRR} \geq (1 + \epsilon) \) then the connection is not feasible.

The first part of the proposition implies that the algorithm finds an \( \epsilon \)-optimal solution with respect to the consumed relative rate, i.e. \( \hat{MRR} < (1 + \epsilon) \cdot MRR^* \).

The second part implies that, when \( 1 < \hat{MRR} < (1 + \epsilon) \), the algorithm identifies an \( \epsilon \)-feasible path, i.e. \( D(\hat{p}) \leq (1 + \epsilon) \cdot D \).

Proof. Let \( j^* \) be the rate class of the maximal guaranteed rate along \( p^* \), i.e., \((1 + \epsilon)j^* \cdot R^1 \leq r(p^*) < (1 + \epsilon)^{j^*+1} \cdot R^1 \). Also, let \( k^* \) be the rate class of the minimal consumed rate \( r_{min}(p^*) \). Denote \( n^* = n(p^*) \). Consider the following two cases:

1. \( k^* < j^* \). Thus, \( MRR^* < \frac{1}{(1+\epsilon)} \).

   Let \( p(n^*, j^*) \) be an \( n^* \)-hops path identified by the algorithm at the \( j^* \)-th iteration. Clearly,

   \[
   r_{min}(p(n^*, j^*)) = \frac{\sigma + n^* \cdot L}{D - \sum_{l \in p(n^*, j^*)} d_l} \leq r_{min}(p^*),
   \]

   and

   \[
   \hat{MRR} \leq MRR(p(n^*, j^*)) = \frac{r_{min}(p(n^*, j^*))}{(1 + \epsilon)^j \cdot R^1} < 1.
   \]

   Furthermore,

   \[
   \frac{r_{min}(p(n^*, j^*))}{(1 + \epsilon)^j \cdot R^1} = \frac{(1 + \epsilon) \cdot r_{min}(p(n^*, j^*))}{(1 + \epsilon)^{j^*+1} \cdot R^1}
   < \frac{(1 + \epsilon) \cdot r_{min}(p(n^*, j^*))}{r(p^*)}
   < \frac{(1 + \epsilon) \cdot r_{min}(p^*)}{r(p^*)}
   \]

   therefore,

   \[
   \hat{MRR} < (1 + \epsilon) \cdot MRR^*.
   \]
2. $k^* = j^*$. Thus, $\frac{1}{(1+\epsilon)} \leq MRR^* \leq 1$.

Let $p(n^*, j^*)$ be a $n^*$-hops path identified by the algorithm at the $j^*$-th iteration. Clearly by the algorithm,

$$r_{\min}(p(n^*, j^*)) \leq r_{\min}(p^*) .$$

For this case,

$$MRR(p(n^*, j^*)) = \frac{r_{\min}(p(n^*, j^*))}{(1+\epsilon)^{j^*} \cdot R^1} \geq 1,$$

but,

$$MRR(p(n^*, j^*)) < (1 + \epsilon) \cdot MRR^* .$$

Since, $\tilde{MRR} \leq MRR(p(n^*, j^*))$, we get

$$\tilde{MRR} < (1 + \epsilon) \cdot MRR^* .$$

If $\tilde{MRR} \leq 1$ then the path $\tilde{p}$ is feasible. Otherwise, If $\tilde{MRR} = \frac{r_{\min}(\tilde{p})}{(1+\epsilon)^{j^*} \cdot R^1} \geq 1$ and $\frac{r_{\min}(\tilde{p})}{r(\tilde{p})} \leq 1$ then $\tilde{p}$ is feasible, however, if $\frac{r_{\min}(\tilde{p})}{r(\tilde{p})} > 1$ then $\tilde{p}$ is not feasible. Whenever $\tilde{p}$ is not feasible the following inequality holds,

$$(1 + \epsilon) \cdot D = (1 + \epsilon) \cdot \left( \frac{\sigma + \tilde{n} \cdot L}{r_{\min}(\tilde{p})} + \sum_{l \in \tilde{p}} d_l \right) \geq \frac{(1 + \epsilon) \cdot (\sigma + \tilde{n} \cdot L)}{(1 + \epsilon) \cdot (1 + \epsilon)^{j^*} \cdot R^1} + \sum_{l \in \tilde{p}} d_l \quad (2.6)$$

$$= D(\tilde{p})$$

If $\tilde{MRR} \geq (1 + \epsilon)$ then $MRR^* > \frac{\tilde{MRR}}{(1+\epsilon)} \geq 1$, i.e., the connection is not feasible.

2.3.3 Variable-Rate Links

Till now, we have assumed that all servers in the network operate at constant rates. However, in practice, the link output rate can be time-varying. Some typical examples are shared-media links governed by MAC protocols, flow controlled links, and mobile links. Also, rate fluctuations can model inaccuracies in the available information regarding the link rate. Inaccuracies can result, for example, from old updates, or from aggregated information in hierarchical networks [8].
Let $S(t)$ denote the instantaneous output transmission capacity of a variable-rate link. Then, the server is said to be fluctuation constraint (FC) with parameters $(\Delta, r)$, if for all $s, \tau$ ( $\tau > s > 0$):

$$S(s, \tau) = \int_s^\tau S(t) \, dt \geq [r(\tau - s) - \Delta]^+$$

(2.7)

where $r$ is the long term average service rate, and $\Delta$ denotes the rate fluctuation.

We apply the results of [20] for a single-input FCFS variable-rate server in isolation, to establish bounds on the delay and backlog for a single RPPS GPS server in isolation. Then, we establish bounds on the end-to-end delay and backlog for an RPPS GPS network. Finally, we incorporate non-negligible packet sizes and deterministic propagation delays.

Consider first a single variable-rate GPS server in isolation.

**Proposition 2.3** Let $\mathcal{I} = \{1, 2, \ldots, I\}$ be the set of sessions entering an RPPS GPS server with a variable service rate. Let the service rate be FC with parameters $(\Delta, R)$, and let the input traffic be BC. For any session $i \in \mathcal{I}$, the backlog and delay are bounded as follows:

$$Q^i(t) \leq \sigma^i + \rho^i \cdot \frac{\Delta}{R},$$

(2.8)

$$D^i(t) \leq \frac{\sigma^i}{r^i} + \frac{\Delta}{R},$$

(2.9)

where $r^i \triangleq \frac{\rho^i}{\sum_{j=1}^n \rho^j} \cdot R$.

**Proof.** We take the decomposition approach [38] in order to decouple the sessions. Accordingly, for an RPPS GPS system, each virtual decomposed server is assigned a guaranteed service of $S^i(\tau, t) = \frac{\rho^i}{\sum_{j=1}^n \rho^j} \cdot S(\tau, t)$. Clearly, the guaranteed service at each decomposed server is FC with the parameters $\left(\frac{r^i}{R} \cdot \Delta, r^i\right)$. Applying the results of Theorem 3.3 and Theorem 3.6 in [20] for a single-input FCFS server concludes the proof.

Next, consider a network with variable-rate RPPS GPS servers. Proposition 2.4 implies that the end-to-end delay in a network with variable-rate RPPS GPS servers equals the delay imposed by a single FC server with parameters $\left(\sum_{n=1}^n \frac{r^{(n)}}{R^{(n)}} \cdot \Delta^{(n)}; r^i(p)\right)$. This result is similar to the one obtained with constant rate links.
Proposition 2.4 For every session $i$ with BC traffic in an RPPS GPS network with FC service rate, we have:

$$Q_{p}^{i}(t) \leq \sigma^{i} + \frac{\rho^{i}}{r^{i}(p)} \cdot \sum_{n=1}^{n(p(i))} \frac{r^{i}_{n}}{R_{n}} \cdot \Delta_{(n)}, \quad (2.10)$$

$$D_{p}^{i}(t) \leq \frac{\sigma^{i}}{r^{i}(p)} + \frac{1}{r^{i}(p)} \cdot \sum_{n=1}^{n(p(i))} \frac{r^{i}_{n}}{R_{n}} \cdot \Delta_{(n)}. \quad (2.11)$$

Proof. See Section 2.6.1.

When packet sizes ($L$) and propagation delays ($d_{l}$) are not negligible, the following proposition holds.

Proposition 2.5 For each session $i$:

$$D(p, r) \leq \frac{\sigma}{r} + \frac{n(p) \cdot L}{r} + \sum_{l \in p} (d_{l} + \Delta_{l}). \quad (2.12)$$

Proof. The end-to-end delay bound, given in (2.12), is derived from Proposition 2.4 by following the same steps as in Section X in [3].

Routing algorithms

A key observation in the design of corresponding QoS routing schemes is that the constant delay $\Delta_{l}$ at each link, caused by the rate fluctuation effect, can be accumulated with the propagation delays. Clearly, the routing schemes introduced in [9] and in Section 2.3 can be applied to variable-rate links as well.

2.4 Stochastic setting

2.4.1 Stochastic Guarantees - EBB Traffic

In this section, we adopt the model introduced in [13] of Exponentially Bounded Burstiness (EBB) processes, defined as follows.

First, we extend the results of [6] to a Packetized-GPS network, and, in addition, incorporate deterministic propagation delays. With the extended upper bound on the end-to-end delay tail distribution at hand, we study the corresponding QoS routing problem. We propose several routing schemes for identifying optimal path with respect to the following criteria: (i) minimizing the end-to-end delay tail distribution; (ii) minimizing a general cost function; (iii) finding a "quickest path". We then consider the application of a rate quantization method, which identifies a near-optimal solution with lower complexity.
Upper Bound on the End-to-End Delay Tail Distribution

Consider a Rate Proportional Processor Sharing GPS network. A session $i$ is routed through a path $p$. The traffic source for session $i$ is modelled as an EBB process. For now, assume that packets are infinitely divisible (fluid model) and that propagation delays are negligible, i.e., the end-to-end delay is solely the outcome of the delays in the queues and processing time. It was shown in [6] that the end-to-end delay tail distribution is bounded by:

$$\Pr \{ D^i_p (t) \geq D \} \leq \Lambda^i_p e^{-\alpha^i \cdot g^i_p \cdot D},$$

where

$$\Lambda^i_p = \Lambda^i \cdot e^{\alpha^i \cdot \rho^i \cdot \xi} \cdot \frac{\ln (\Lambda^i + 1)}{\alpha^i \cdot (g^i_p - \rho^i)}, \quad 0 < \xi < \frac{\ln (\Lambda^i + 1)}{\alpha^i \cdot (g^i_p - \rho^i)},$$

and $g^i_p \triangleq \min_{l \in p} r^i_l$ is the guaranteed backlog clearing rate for session $i$ at the bottleneck.

Therefore, the minimal bound is achieved when:

$$\Lambda^i_p = \begin{cases} \Lambda^i \cdot \frac{g^i_p}{g^i_p - \rho^i} \cdot \frac{(g^i_p - \rho^i)^{\alpha^i \cdot \rho^i \cdot \xi}}{\rho^i}, & g^i_p \leq \rho^i (\Lambda^i + 1) \\ (\Lambda^i + 1)^{\frac{\alpha^i \cdot \rho^i \cdot \xi}{g^i_p - \rho^i}}, & g^i_p > \rho^i (\Lambda^i + 1) \end{cases} \quad (2.13)$$

Packetized Traffic  We extend now the results of [6] to Packetized traffic and establish upper bounds on the end-to-end delay tail distribution for a PGPS network.

Assume that the traffic flow that each session $i$ introduces to the network is composed of packets, and denote the maximal packet size of a session $i$ by $L^i$. When packet size is not negligible, there are two effects to be considered [3]. First, packets are served non-preemptively, i.e., once the server has begun serving a packet, it continues to do so until completion. Therefore, for each single server in the route [2]:

$$\tilde{Q}^i (0, \tau) - Q^i (0, \tau) \leq L_{\text{max}},$$

where $\tilde{Q}^i (0, \tau)$ and $Q^i (0, \tau)$ are the total backlogs for session $i$ at time $\tau$ for a non-preemptive PGPS server and a GPS server, respectively, and $L_{\text{max}}$ is the maximal packet size of all sessions. Since the packet is served at rate $R_{(n)}$,

$$\tilde{D}_p^i (t) \leq D_p^i (t) + \sum_{n=1}^{n(p)} \frac{L_{\text{max}}}{R_{(n)}}, \quad (2.14)$$

30
where \( \tilde{D}_i^p(t) \) and \( D_i^p(t) \) are the end-to-end delays for session \( i \) at time \( t \) for a non-preemptive PGPS network and a GPS network, respectively.

Second, the service is *non-cut-through*, i.e., no packet is eligible for service until its last bit has arrived. Hence, we cannot assume that \( S_{(n-1)}^i = A_{(n)}^i \), but rather:

\[
S_{(n-1)}^i (\tau, t) + L^i \geq A_{(n)}^i (\tau, t) - L^i \quad n = 2, \ldots, n(p), \quad \tau < t \quad (2.15)
\]

With the above observations, we can establish the following bounds for packetized traffic.

**Proposition 2.6** For every session \( i \) with EBB traffic in an RPPS PGPS network, and for any \( D > 0 \):

\[
\Pr \left\{ \tilde{D}_i^p(t) \geq D \right\} \leq \Lambda_p e^{-\alpha} \left( g_p^i \left( D - \sum_{n=1}^{n(p)} \frac{L_{\text{max}}}{R_{(n)}} \right) - n(p) \cdot L^i \right). \quad (2.16)
\]

*Proof.* See Section 2.6.2.

**Propagation Delays** Suppose that each bit transmitted on a link \( l \) incurs a constant delay of \( d_l \) units. It is easy to see that:

\[
\tilde{D}_i^p(t) = D_i^p(t) + \sum_{l \in p} d_l,
\]

and the corresponding bound is:

\[
\Pr \left\{ \tilde{D}_i^p(t) \geq D \right\} \leq \Lambda_p e^{-\alpha} \left( g_p^i \left( D - \sum_{l \in p} d_l - \sum_{n=1}^{n(p)} \frac{L_{\text{max}}}{R_{(n)}} \right) - n(p) \cdot L^i \right). \quad (2.17)
\]

For broadband networks, we may assume that \( R_{(n)} \to \infty \), so the term \( \sum_{n=1}^{n(p)} \frac{L_{\text{max}}}{R_{(n)}} \) can be neglected. Alternatively, the constant delay \( \frac{L_{\text{max}}}{R_{(n)}} \) at each queue, caused by the non-preemptive effect, can be accumulated with the propagation delays.

Accordingly, when a session \( i \) with EBB traffic is routed over a path \( p \) in an RPPS GPS network with a reserved rate \( r \) (where \( r \leq r(p) \)), for any \( D > 0 \), the end-to-end delay tail distribution is upper bounded as follows:

\[
\Pr \{ D(p, r) \geq D \} \leq \begin{cases} 
\Lambda \left( \frac{r}{r(p)} \right) \left( \frac{x}{x(p)} \right)^{\frac{r}{r(p)}} \cdot e^{-a} \left( r \left( D - \sum_{l \in p} d_l \right) - n(p) \cdot L^i \right) & r \leq \rho(\Lambda + 1) \\
(\Lambda + 1)^{\frac{r}{r(p)}} \cdot e^{-a} \left( r \left( D - \sum_{l \in p} d_l \right) - n(p) \cdot L^i \right) & r > \rho(\Lambda + 1)
\end{cases} \quad (2.18)
\]
The bound in (2.18) follows from (2.17), since $q^p > r$ (where $r \leq r(p)$). For ease of presentation, we omitted the session index $i$. Denote this end-to-end delay tail distribution upper bound by $Pr(D(p, r))$. We shall also denote by $Pr(D, p)$ the minimal possible value of $Pr(D(p, r))$, i.e., $Pr(D, p) = Pr(D(p, r(p)))$.

**Routing Algorithms**

With the above upper bound (2.18) on the end-to-end delay tail distribution at hand, we investigate the corresponding routing problem. Obviously, the routing schemes introduced under the deterministic setting (e.g. [8, 9, 10, 24, 25]) cannot be immediately applied, and new schemes are called for.

**Path with Minimal End-to-End Delay Tail Distribution**  We begin by considering the basic problem of identifying $q$-feasible paths. If several $q$-feasible paths exist, we seek a path with the minimal end-to-end delay tail distribution. The problem, then, is to find a path that minimizes the delay tail distribution upper bound, i.e., solve the following problem:

$$
\min_p Pr(D, p) = \begin{cases} 
\Lambda \left( \frac{r(p)}{r(p)^{-\rho}} \right) \cdot e^{-\alpha \left( r(p) \left( D - \sum_{l \in \mathcal{P}} d_l \right) - n(p) \cdot L \right)} & r(p) \leq \rho (\Lambda + 1) \\
(\Lambda + 1) \left( \frac{r(p)}{r(p)^{-\rho}} \right) \cdot e^{-\alpha \left( r(p) \left( D - \sum_{l \in \mathcal{P}} d_l \right) - n(p) \cdot L \right)} & r(p) > \rho (\Lambda + 1)
\end{cases}
$$

Algorithm Minimum Delay tail Distribution, depicted in Figure 2.3, identifies such a path.

**Proposition 2.7**  (i) Algorithm MDD correctly identifies the $q$-feasibility of the connection.

(ii) Whenever the connection is $q$-feasible, the path $p(\hat{k})$, identified by the algorithm, achieves the minimal delay tail distribution upper bound $Pr(D, p(\hat{k}))$ among all $q$-feasible solutions.

(iii) The algorithm’s complexity is $O((N \log N + M) \cdot K)$.

*Proof.* The first and third parts of the proposition are straightforward. Consider, then, a $q$-feasible connection. Recall that $R^1 \leq R^2 \leq \ldots \leq R^K$ are the different values of $R_l^i$, where $K \leq M$. Let $p^*$ be the path that minimizes the bound among all $q$-feasible solutions, and $k^*$ the index for which $R^{k^*} = r(p^*)$. Consider the path...
1. for \( k ← 1 \) to \( K \)
   (a) delete all links \( l \) with \( R'_l < R^k \)
   (b) find a path \( p(k) \) that is shortest with respect to the metric \( \{ \frac{L}{R^k} + d_l \} \) through Dijkstra’s shortest path algorithm
   (c) taking \( r(p(k)) ← R^k \), compute \( \Pr(D, p(k)) \)

2. choose the path \( p(\tilde{k}) \) with the minimal upper bound \( \Pr(D, p(\tilde{k})) \) among the \( K \) paths, \( p(k) \ 1 ≤ k ≤ K \)

3. if \( \Pr(D, p(\tilde{k})) \leq q \) then \( p(\tilde{k}) \) is \( q \)-feasible path, else there is no \( q \)-feasible path

Figure 2.3: Algorithm Minimum Delay tail Distribution (MDD)

\( p(k^*) \), identified at the \( k^* \)-th iteration of the algorithm. We have: \( r(p(k^*)) \geq R^{k^*} \), therefore,

\[
r(p^*) \leq r(p(k^*)) .
\]

(2.20)

Clearly, by the algorithm,

\[
\sum_{l \in p(k^*)} \left\{ \frac{L}{R^{k^*}} + d_l \right\} \leq \sum_{l \in p} \left\{ \frac{L}{R^{k^*}} + d_l \right\} .
\]

(2.21)

Next, we establish that

\[
\Lambda_{(p(k^*))} \leq \Lambda_{(p^*)},
\]

(2.22)

where \( \Lambda_{(p)} \) is given in (2.13).

To that end, we show that \( \Lambda_{(p)} (r) \) is a non-decreasing function for all \( r > \rho \), where

\[
\Lambda_{(p)} (r) = \begin{cases} 
\Lambda \left( \frac{r}{r-\rho} \right) \cdot \left( \frac{\rho}{r} \right)^{r-\rho} & r \leq \rho \ (\Lambda + 1) \\
(\Lambda + 1) & r > \rho \ (\Lambda + 1)
\end{cases} .
\]

(2.23)

First, consider the case \( r \leq \rho \ (\Lambda + 1) \). Let \( f(r) = \frac{r \ln(r) - (r-\rho) \ln(r-\rho) - \rho \ln(\rho)}{r-\rho} \). It is easy to see that \( \frac{df(r)}{dr} = \frac{\rho \ln(\rho) - \ln(r)}{(r-\rho)^2} < 0 \ \forall r > \rho \), thus \( f(r) \) is a non-decreasing function for all \( r > \rho \). It is easy to verify that \( f(r) = \ln \left( \frac{r}{r-\rho} \right) + \frac{\rho}{r-\rho} \ln \left( \frac{\rho}{r} \right) \) and that \( \left( \frac{r}{r-\rho} \right) \left( \frac{\rho}{r} \right)^{\frac{\rho}{r-\rho}} = e^{f(r)} \). Thus, \( \Lambda \left( \frac{r}{r-\rho} \right) \left( \frac{\rho}{r} \right)^{\frac{\rho}{r-\rho}} \) is non-decreasing for all \( r > \rho \).

Next, considering the case \( r > \rho \ (\Lambda + 1) \), one can see that \( (\Lambda + 1)^{\frac{r}{r-\rho}} \) is non-decreasing in this case too.
Finally, it is easy to see that $\Lambda_p(r)$ is continuous at $r = \rho(A + 1)$. Thus, $\Lambda_p(r)$ is a non-decreasing function for all $r > \rho$ and (2.22) follows immediately from (2.20).

Combining (2.19), (2.20), (2.21) and (2.22), we get:

$$\Pr(D_\rho(k^*)) \leq \Pr(D_\rho(p^*)) .$$

Finally, from the algorithm, $\Pr(D, p(\hat{k})) \leq \Pr(D, p(k^*))$, thus the path $p(\hat{k})$ identified by the algorithm achieves the minimal upper bound.

**Minimum Cost Path** Next, we consider the problem of finding a feasible path that optimizes some general cost function. Following [9], we consider a cost function $C(r, p)$, which depends on the consumed rate $r$ and the path $p$. The only assumption that we make is that $C(r, p) = C(r, n(p), \bar{D}(p, r, q))$ and that it is non-decreasing in each of its three arguments. In other words, the cost is a non-decreasing function of (i) the number of hops $n(p)$, (ii) the consumed rate $r$, and (iii) the effective end-to-end delay $\bar{D}(p, r, q)$ (where, $\Pr(D(p, r) \geq \bar{D}(p, r, q)) \leq q$).

Algorithm Minimum Cost, depicted in Figure 2.4, identifies an optimal (i.e., minimum cost) path.

1. for $k \leftarrow 1$ to $K$
   
   (a) delete all links $l$ with $R'_l < R^k$
   
   (b) for $n \leftarrow 1$ to $H$
   
   i. find a path $p(n, k)$ that is the shortest with respect to the metric $\{d_l\}$ among $n$-hops paths.
   
   ii. if $\bar{D}(p, q) \leq D$ then
   
   A. $r(n, k) \leftarrow \arg \min_{r \leq R^k} C(r, p(n, k))$
   
   B. $MC(n, k) \leftarrow C(r(n, k), p(n, k))$
   
   iii. else $MC(n, k) \leftarrow \infty$

2. let $\hat{n}$ and $\hat{k}$ be such that $MC(\hat{n}, \hat{k})$ is minimal over all $1 \leq n \leq H$ and $1 \leq k \leq K$. If $MC(\hat{n}, \hat{k}) = \infty$ then the connection is not $q$-feasible, otherwise $p(\hat{n}, \hat{k})$ and $r(\hat{n}, \hat{k})$ are the required path and rate, correspondingly.

**Figure 2.4: Algorithm Minimum Cost (MC)**

**Proposition 2.8** Algorithm $MC$ correctly identifies the $q$-feasibility of the connection. Whenever the connection is $q$-feasible, the path $p(\hat{n}, \hat{k})$ and rate $r(\hat{n}, \hat{k})$,
identified by the algorithm, achieve the minimal cost among all \( q \)-feasible solutions. The algorithm’s complexity is \( O(M \cdot H \cdot K) \).

**Proof.** The proof of the proposition is similar to the proof of Proposition 5 in [9], and is specified in Section 2.6.3 for completeness. ■

**Quickest Path** The third routing scheme seeks a "quickest path", *i.e.*, a path with a minimal effective end-to-end delay. The problem, then, is to find a path that minimizes the following expression:

\[
\bar{D}(p, q) = \left\{ \begin{array}{ll}
\sum_{l \in P} d_l + \frac{n(p) L}{r(p)} - \frac{\ln \left( \frac{q}{r(p)} \right)}{\alpha r(p)} + \frac{\ln \left( \frac{\rho L}{\alpha r(p) \left( r(p) - \rho \right)} \right)}{\alpha r(p) \left( r(p) - \rho \right)} & r(p) \leq \rho ( \Lambda + 1 ) \\
\sum_{l \in P} d_l + \frac{n(p) L}{r(p)} - \frac{\ln(\Lambda + 1)}{\alpha r(p)} & r(p) > \rho ( \Lambda + 1 )
\end{array} \right.
\]

(2.24)

Such a path can be identified by the execution of the MC algorithm with \( C(r, p) = \bar{D}(p, q) \). Alternatively it can be identified through \( K \) executions of Dijkstra’s shortest path algorithm with respect to the metric \( \{ \frac{L}{\hat{R}} + d_l \} \). Thus, the routing problem can be solved with complexity of \( O(M \cdot H \cdot K) \) or \( O((N \log N + M) K) \). While being polynomial, such complexity could still be prohibitive [9].

Hence, we propose a rate quantized scheme, which identifies a near-optimal solution with lower complexity. Recall that the available rate values in the network are denoted by \( R^1, R^2, \ldots, R^K \), where \( K \leq M \), and let \( \bar{R} = \frac{R^K}{R^1} \). As in [9], our aim is to establish efficient \( \epsilon \)-optimal routing schemes. There, given a value \( \epsilon > 0 \), the available rate values are grouped into \( O \left( \log_{1+\epsilon} \bar{R} \right) \) rate classes, such that for \( 0 \leq j \leq \lfloor \log_{1+\epsilon} \bar{R} \rfloor \), rate class \( j \) contains all maximal rates in the range \( R^1 (1 + \epsilon)^j \ldots R^1 (1 + \epsilon)^{j+1} \). A link \( l \) is in rate-class \( j \) if \( R^1 (1 + \epsilon)^j \leq R^l_1 < R^1 (1 + \epsilon)^{j+1} \). Applying such a standard (uniform) rate quantization method leads to an \( \epsilon \)-optimal solution with complexity of \( O \left( (N \log N + M) \cdot \min \left\{ \frac{2R^1 - \rho}{(R^1 - \rho) \epsilon} \cdot \log \bar{R}, K \right\} \right) \). Since \( (R^1 - \rho) \) may be small, the complexity of this rate quantized scheme could still be prohibitive. Thus, a *non-uniform* rate quantized scheme is required. With such a scheme, the rate-classes are determined according to the required approximation. Accordingly, we group the available rate values in the network into non-uniform *rate-classes*. We say that a link \( l \) is in rate-class \( j \) if \( R(j) \leq R^l_1 < R(j + 1) \), where \( R(0) = R^1 \), \( R(j + 1) = a_{j+1} \cdot R(j) \) and \( R^{K-1} \leq R(J) \leq R^K \). The parameters \( a_j \), \( 1 \leq j \leq J \), are set to assure an \( \epsilon \)-optimal solution. The calculation of these values is done (in step 2e) following the
formulation presented in the proof of Proposition 2.9 that follows.

The Rate-Quantized Quickest path algorithm is formulated in Figure 2.5.

1. $R(0) \leftarrow R^1$

2. for $j \leftarrow 0$ to $J$
   
   (a) if rate-class $j$ is empty then skip to the next value of $j$
   
   (b) delete from the network all links whose rate-class is less than $j$
   
   (c) find the shortest path $p(j)$ with respect to the metric \( \left\{ \frac{L}{R(j)} + d_l \right\} \)
       through Dijkstra’s shortest path algorithm
   
   (d) compute $\bar{D}(p(j), q)$
   
   (e) if $R(j) < \rho(\Lambda + 1)$ then

$$a_{j+1} \leftarrow \frac{\rho + (1 + \epsilon)^{\frac{j+1}{2}} (R^1 - \rho)}{\rho + (1 + \epsilon)^{\frac{j}{2}} (R^1 - \rho)}$$

else

$$a_{j+1} \leftarrow \frac{1 + (1 + \epsilon)^{\frac{j+1}{2}} \Lambda}{1 + (1 + \epsilon)^{\frac{j}{2}} \Lambda}$$

(f) $R(j + 1) \leftarrow a_{j+1} \cdot R(j)$

3. among all paths $p(j)$, choose a path $p(\tilde{j})$ with the minimal effective delay $\bar{D}(p(\tilde{j}), q)$

Figure 2.5: Algorithm Quickest Path - Rate Quantized (QP-RQ)

**Proposition 2.9** The effective end-to-end delay of the path $p(\tilde{j})$, identified by algorithm QP-RQ, is at most $1 + \epsilon$ times larger than the minimal value, i.e., if $p^*$ is the optimal path then

$$\bar{D}(p(\tilde{j}), q) \leq (1 + \epsilon) \cdot \bar{D}(p^*, q).$$

The algorithm’s complexity is

$$O \left( (N \log N + M) \cdot \min \left\{ \frac{1}{\epsilon} \log \frac{\rho \Lambda (R^K - \rho)}{(R^1 - \rho)}, K \right\} \right).$$

**Proof.** See Section 2.6.4.

One can see that, with non-uniform quantization, the complexity is much lower, since now the complexity is relative to the logarithm of the expression $\frac{1}{(R^1 - \rho)}$.  

36
2.4.2 Simulation results

We now demonstrate the efficiency of the routing algorithms proposed in Section 2.4.1 by way of simulations. First, we compare the stochastic minimum delay tail distribution (MDD) routing algorithm with the deterministic single-commodity feasible path (SFP) algorithm of [9]. Through a simple network example, we indicate that the deterministic SFP algorithm may identify a false route, which is not \( q \)-feasible; in contrast, our MDD algorithm is guaranteed to correctly identify a \( q \)-feasible route (when one exists, as in this case). Next, we illustrate the advantage of the stochastic minimum cost algorithm through comparison with two standard QoS routing algorithms, namely, the “shortest-widest” and “widest-shortest” routing schemes [36].

Consider some source and destination nodes connected by two parallel links \( l_1 \) and \( l_2 \), as depicted in Figure 2.6. The bandwidth of both links is assumed to be 1.

We consider the discrete time case and employ the following discrete version of the delay tail distribution bound:

\[
\Pr\{D(p) \geq D\} \leq \frac{\Lambda}{1-e^{-\alpha(r-\rho)}} \cdot e^{-\alpha \left(r \left(D - \sum_{i \in p} d_i \right) - n(p) \cdot L \right)},
\]

This bound is obtained by applying the discrete time version of the bound (2.13) and by following the same steps as in Section 2.4.1.

Suppose that three session are established and assume that the first session is routed through link \( l_1 \), the second session is routed through link \( l_2 \), and the third session can be routed through either \( l_1 \) or \( l_2 \). We assume that the source traffic for each session is modelled by a (mutually) independent Bernoulli process, which is known to be EBB [13]. The sessions parameters are presented in Table 2.1.

We consider two routing strategies: (i) our minimum delay tail distribution (MDD) algorithm and (ii) the deterministic single-commodity feasible path (SFP) algorithm of [9]. For the deterministic SFP algorithm, we neglect bursts of probability less than \( q \), thus, we take the deterministic maximal burst length to be \( \sigma = \frac{1}{\alpha} \ln \left( \frac{\Lambda}{q} \right) \). Under the above settings, the stochastic MDD algorithm identifies
link $l_1$ as the optimal ($q$-feasible) path, whereas the deterministic SFP identifies link $l_2$. In Figure 2.7 we plot the bounds of the delay tail distribution of session 3 (solid lines) as well as the actual simulation results (dotted lines) for the two scenarios: (i) session 3 is routed through $l_1$ and (ii) session 3 is routed through $l_2$. One can see that $l_1$, which was identified by SFP, is not a $q$-feasible path (i.e., $\Pr\{D_3(l_1) > 14\} > 10^{-4}$) whereas $l_2$, which was identified by MDD, is indeed a $q$-feasible path, as required.

![Network topologies](figure.png)

(a) MCI topology  
(b) Clustered topology

Figure 2.8: Networks topologies

Next, we illustrate the advantage of our minimum cost routing scheme in
balancing the load. We set the cost function to be the consumed rate, \( i.e., C(r,p) = r \cdot n(p) \). Our figure of merit is session acceptance probability, which is evaluated for various loads and network topologies. We assume that each session has EBB traffic with rate \( \rho = 1 \text{Mbps}, \Lambda = 1 \) and \( \alpha = 2 \). Furthermore, we assume that each session requires a maximum end-to-end effective delay of 2msec and \( q = 10^{-4} \). We also assume that the session’s traffic is packetized and that the packet sizes are according to a trimodal distribution, namely, 60% of the packets being 44 bytes, 20% of the packets 552 bytes and the rest 1500 bytes [39]. Following [10], we consider the two network topologies depicted in Figure 2.8. Session requests arrive in sequentially. Once resources are reserved within the network, they are held for the entire duration of the simulation experiment. We compare the following three routing algorithms:

- Our algorithm Minimum-Rate, which identifies a \( q \)-feasible path with the minimum consumed rate, \( i.e., \) algorithm Minimum-Cost with \( C(r,p) = r \cdot n(p) \).

- The “widest-shortest” algorithm [36], which, in our setting, identifies a \( q \)-feasible path with the minimum hop count. If there are several such paths, the one with the maximum residual bandwidth is selected.

- The “shortest-widest” algorithm [36], which, in our setting, identifies a \( q \)-feasible path with the maximum residual bandwidth. If there are several such paths, the one with the minimum hop count is selected.

\[
\text{Figure 2.9: Flow request blocking probability - evenly distributed load}
\]

Figure 2.9 presents the blocking probability as a function of the load (\( i.e., \) number of session requests) for the two network topologies when the traffic is evenly distributed (source and destination nodes are randomly selected). Note that the blocking rate is evaluated as the average number of rejected sessions divided by the
number of session requests. Figure 2.10 shows the results for an uneven load, where most of the traffic (90%) is between “East” and “West” (the source/destination node is randomly selected from the West/East nodes, respectively). In practice, blocking rates higher than some 10% may be prohibitive. Hence, focusing on the load region for which the blocking rate is lower than 10%, our minimum rate algorithm outperforms both the widest-shortest as well as the shortest-widest algorithms in all considered scenarios. Furthermore, considering all traffic loads (i.e., also those resulting with block rates of more than 10%), our minimum-rate algorithm achieves good network utilization both for evenly as well as unevenly distributed traffic; on the other hand, the shortest-widest and widest-shortest algorithms achieve bad network utilization for evenly/unevenly distributed traffic, respectively.

2.4.3 Stochastic Guarantees - SBB Traffic

We consider a more general source traffic model, of \textit{Stochastically Bounded Burstiness (SBB)} processes, whose burstiness is stochastically bounded by a general decreasing function [14]. The SBB approach is based on a generalization of the EBB network calculus, where only exponentially decaying bounding functions were considered. This approach has two major advantages: (i) it applies to a larger class of input processes, and (ii) it provides much better bounds for common models of real-time traffic. [14] formulated the SBB calculus for an isolated network element and considered the stability of a feed-forward network. We proceed to present the formal definition.

We consider SBB traffic entering an RPPS PGPS network, and establish bounds on the end-to-end backlog and delay tail distribution. We then consider the related routing problem; due to the complexity of the bounds, we focus on the special case in which the bounding function is the sum of two exponents, and provide a
near-optimal routing algorithm of low complexity.

Upper Bound on the End-to-End Delay Tail Distribution

We begin by considering a single RPPS GPS server and an SBB input traffic stream.

**Proposition 2.10** Let \( A^i(t) \) be the session \( i \) input traffic stream to an RPPS GPS server. If \( A^i(t) \) is SBB with upper rate \( \rho^i < r^i \) and bounding function \( f(\sigma^i) \), then for any \( \xi > 0 \),

\[
\Pr \{ Q^i(t) \geq Q \} \leq f\left( (Q - \rho^i \cdot \xi)^+ \right) + \frac{1}{(r^i - \rho^i) \cdot \xi} \int_{Q - \rho^i \cdot \xi}^{\infty} f(u) \, du, \tag{2.26}
\]

and

\[
\Pr \{ D^i(t) \geq D \} \leq f\left( (r^i \cdot D - \rho^i \cdot \xi)^+ \right) + \frac{1}{(r^i - \rho^i) \cdot \xi} \int_{r^i \cdot D - \rho^i \cdot \xi}^{\infty} f(u) \, du, \tag{2.27}
\]

where the notation \((\cdot)^+\) denotes the operation \(\max(\cdot, 0)\).

**Proof.** We apply the decomposition approach in the analysis of a GPS server [38]. Accordingly, each virtual decomposed server has a single input. The proposition follows immediately from the results of [14] for an isolated network element with SBB input traffic.

Next, we establish bounds on the end-to-end delay tail distribution in an RPPS PGPS network.

**Proposition 2.11** For every session \( i \) with SBB traffic in an RPPS PGPS network, for any \( \xi > 0 \), \( Q > 0 \) and \( D > 0 \):

\[
\Pr \{ Q^i_p(t) \geq Q \} \leq f\left( (Q - n(p) \cdot L^i - \rho^i \cdot \xi)^+ \right) + \frac{1}{(r(p) - \rho^i) \xi} \int_{Q - n(p) \cdot L^i - \rho^i \cdot \xi}^{\infty} f(u) \, du, \tag{2.28}
\]

\[
\Pr \{ D^i_p(t) \geq D \} \leq f\left( (\eta^i(p) - \rho^i \cdot \xi)^+ \right) + \frac{1}{(r(p) - \rho^i) \xi} \int_{\eta^i(p) - \rho^i \cdot \xi}^{\infty} f(u) \, du, \tag{2.29}
\]

where, \( \eta^i(p) \triangleq r(p) \left( D - \sum_{l \in p} d_l \right) - n(p) \cdot L^i \).

**Proof.** See Section 2.6.5.
Special Case: Sum of Exponents

As a special case of the SBB model, consider a bounding function that is the sum of two exponents. For this case, we can provide better bounds than those obtained for the EBB model (i.e., a single exponent); at the same time, these bounds have a simple closed form, just as in the EBB case.

Let the bounding function be

\[ f(\sigma) = \Lambda e^{-\alpha \sigma} + Be^{-\beta \sigma}, \]

for all \( \sigma \geq 0 \), \( 0 \leq s \leq \tau \),

\[ \Pr\{ A(s, \tau) \geq \rho (\tau - s) + \sigma \} \leq \Lambda e^{-\alpha \sigma} + B e^{-\beta \sigma}. \]

According to proposition 2.11, in an RPPS PGPS network, for every session \( i \) with SBB traffic and a bounding function \( f(\sigma) = \Lambda e^{-\alpha \sigma} + Be^{-\beta \sigma} \) and for any \( \xi > 0 \) and \( D > 0 \):

\[ \Pr\{ D(p) \geq D \} \leq \left( 1 + \frac{1}{\alpha (r(p) - \rho) \xi} \right) \cdot \Lambda \cdot e^{-\alpha (\eta(p) - \rho \xi)} + \left( 1 + \frac{1}{\beta (r(p) - \rho) \xi} \right) \cdot B \cdot e^{-\beta (\eta(p) - \rho \xi)}, \quad (2.30) \]

where, \( \eta(p) \triangleq r(p) \left( D - \sum_{t \in p} d_t \right) - n(p) \cdot L. \)

For the ease of presentation, we omitted the session index \( i \). Denote the upper bound on end-to-end delay tail distribution, given in (2.30), as \( \overline{\Pr}(D, p) \).

Example

We now illustrate the results we have obtained for RPPS-GPS networks with SBB traffic through an example. Specifically, following [6, 13], we consider the three-node tree structured network depicted in Figure 2.11. The rates of the servers are assumed to be as follows: \( r_1 = 2.5 \), \( r_2 = 2.1 \), and \( r_3 = 4.5 \). Suppose that there are four sessions in the network, which are routed as depicted in Figure 2.11. The source traffic for each session is modelled by a discrete time Markov modulated process \( A(t), t \in \mathbb{N} \), defined as \( A(t) = Y_{B(t)}(t) \). The modulating chain \( B(t) \) is assumed to have two states (say 1,2) with transition probabilities \( p_{12} \) and \( p_{21} \).

When in state 1, the source is producing i.i.d. arrivals distributed according to \( \Pr(Y_1(t) = 0) = 1 - \Pr(Y_1(t) = 2) \). When in state 2, the source is producing i.i.d. arrivals distributed according to \( \Pr(Y_2(t) = 0) = 1 - \Pr(Y_2(t) = 2) \). The Markov modulated process parameters for each session are presented in Table 2.2. Using standard z-transform techniques it can be shown that \( A(t) \) is SBB with upper rate \( \rho = 1 \) and bounding function that is the sum of two exponents. Specifically, sessions 1,3 are SBB with a bounding function \( f(\sigma) = e^{-1.946\sigma} + 10^{-4}e^{-0.273\sigma} \forall \sigma \geq 0 \) and sessions 2,4 are SBB with a bounding function \( f(\sigma) = e^{-2.197\sigma} + 10^{-4}e^{-0.543\sigma} \forall \sigma \geq 0. \)
Choosing the same GPS assignments for all sessions at all nodes, say, \( \phi^i = \rho^i = 1 \) for \( i = 1, 2, 3, 4 \), and applying Proposition 2.11, we obtain bounds on the tail distribution of the end-to-end delay for the four sessions. More precisely, we consider a discrete time version of the bound in (2.30). Using the discrete time case results for an isolated network element with SBB input traffic (Theorem 3 of [14]) and then, following the same steps as in sections 2.4.3 and 2.4.3, we have that, for \( D > 0 \),

\[
\Pr \{ D(p) \geq D \} \leq \left(1 + \frac{1}{\alpha (r(p) - \rho)}\right) \Lambda e^{-\alpha(r(p)D-n(p)L)} + \left(1 + \frac{1}{\beta (r(p) - \rho)}\right) B e^{-\beta(r(p)D-n(p)L)}.
\]

These bounds are presented by the solid lines in Figure 2.12. The simulation results for the actual Markov modulated processes are presented by the dashed lines. In that figure, we also plot (dotted lines) the bounds obtained by the EBB calculus. As can be seen, the SB (i.e., sum of two exponents) bound is much tighter than the EB (i.e., one exponent) bound.

**Routing Algorithms**

We consider the problem of finding a path with the minimal bound on the end-to-end delay tail distribution. Although the bound (2.30) is quite complex, it is easy to verify that it is still the same, relatively simple, metric (for a given \( r \)), namely \( \{L \bar{r} + dl\} \), that characterizes the path with respect to the end-to-end delay tail dis-

---

**Table 2.2: Arrival processes parameters**

<table>
<thead>
<tr>
<th>session</th>
<th>( p_{12} )</th>
<th>( p_{21} )</th>
<th>( \Pr(Y_1 = 0) )</th>
<th>( \Pr(Y_2 = 0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 10^{-6} )</td>
<td>1/100</td>
<td>7/8</td>
<td>11/20</td>
</tr>
<tr>
<td>2</td>
<td>( 10^{-6} )</td>
<td>1/100</td>
<td>9/10</td>
<td>5/8</td>
</tr>
<tr>
<td>3</td>
<td>( 10^{-6} )</td>
<td>1/100</td>
<td>7/8</td>
<td>11/20</td>
</tr>
<tr>
<td>4</td>
<td>( 10^{-6} )</td>
<td>1/100</td>
<td>9/10</td>
<td>5/8</td>
</tr>
</tbody>
</table>
Figure 2.12: SBB network example (- - Simulation; — SB bound; · · · EB bound)

Proposition 2.12 The path \( \tilde{p} \) identified by the algorithm MDD-RQ, constitutes an \( \epsilon \)-optimal solution, i.e.:

\[
\Pr((1+\epsilon) \cdot D, \tilde{p}) \leq \Pr(D, p^*)
\]

The algorithm’s complexity is

\[
O \left( (N \log N + M) \cdot \min \left\{ \frac{1}{\epsilon} \log \left( \frac{R^K - \rho}{R^1 - \rho} \right), K \right\} \right).
\]

Proof. See Section 2.6.6.

2.4.4 Variable-Rate Links

Here, we consider the stochastic fluctuation setting. We derive bounds on the end-to-end delay, and propose corresponding routing algorithms.

Let \( S(t) \) denote the instantaneous output transmission capacity of a variable-rate link. Then, the server is said to have \textit{Exponentially Bounded Fluctuation (EBF)}
Figure 2.13: Algorithm Minimum Delay tail Distribution - Rate Quantized (MDD-RQ)

[20] with parameters \((r, B, \beta)\), if for all \(\tau, t \ (t > \tau > 0)\):

\[
\Pr \left\{ \sum_{n=\tau+1}^{t} S(n) \leq r(t - \tau) - \Delta \right\} \leq B \cdot e^{-\beta \Delta} \tag{2.31}
\]

where \(r\) is the long term average service rate, and \(\Delta\) denotes the rate fluctuation.

Note that the results for variable-rate links, under a stochastic setting, are derived for a discrete time domain.

Consider a single RPPS GPS server with an EBF service rate and EBB input traffic. The following proposition states that the traffic backlog and delay are exponentially bounded (EB).

**Proposition 2.13** Consider a set of \(I\) sessions \(\{1, 2, \ldots, I\}\) with EBB input traffic \((\rho^i, \Lambda^i, \alpha^i)\) entering an RPPS GPS server with a variable service rate. Let the service rate have exponentially bounded fluctuation with parameters \((R, B, \beta)\). For every session \(1 \leq i \leq I\) and for any \(t > 0\),

\[
\Pr \{ Q^i(t) \geq Q \} \leq \frac{\Lambda^i + B}{1 - e^{-\xi^i(g^i - \rho^i)}} e^{-\xi^i \cdot Q}, \tag{2.32}
\]

\[
\Pr \{ D^i(t) \geq D \} \leq \frac{\Lambda^i + B}{1 - e^{-\xi^i(g^i - \rho^i)}} e^{-\xi^i \cdot g^i \cdot D}, \tag{2.33}
\]
where
\[ g^i \triangleq \frac{r^i}{\sum_{j=1}^{r^i} R_j}, \beta^i = \frac{\beta R}{r^i}, \text{ and } \xi^i \triangleq \frac{\alpha^i \beta^i}{\alpha^i + \beta^i}. \]

**Proof.** We take a decomposition approach.

Since the service rate is EBF we have \( \Pr \{ S(\tau, t) \leq R(t - \tau) - \Delta \} \leq B \cdot e^{-\beta \Delta}. \)

With a GPS server we have \( S^i(\tau, t) \geq \frac{r^i}{\sum_{j=1}^{r^i} R_j} \cdot S(\tau, t) \geq \frac{r^i}{R} \cdot S(\tau, t). \)

Therefore,
\[ \Pr \left\{ S^i(\tau, t) \leq \frac{r^i}{R} \cdot R \cdot t - \frac{r^i}{R} \cdot \Delta \right\} \leq B \cdot e^{-\beta \Delta} \]
and
\[ \Pr \{ S(\tau, t) \leq r^i(t - \tau) - \Delta^i \} \leq B \cdot e^{-\frac{\beta R}{r^i} \Delta^i}. \]
That is, the guaranteed service rate at each virtual decomposed server has exponentially bounded fluctuation with parameters \((g^i, B, \beta^i)\). Applying the results of Theorem 4.3 and Theorem 4.5 in [20], for a single-input EBB traffic stream entering a single EBF server, concludes the proof.

Consider now an RPPS GPS network with EBF service rates and EBB input traffic.

**Proposition 2.14** For every session \( i \) with EBB traffic in an RPPS GPS network with EBF service rate, we have:

\[
\Pr \{ Q_p^i(t) \geq Q \} \leq \frac{\Lambda^i + B_p}{1 - e^{-\xi_p^i (g^i_p - \rho^i)}} e^{-\xi_p^i Q},
\]
\[
\Pr \{ D_p^i(t) \geq D \} \leq \frac{\Lambda^i + B_p}{1 - e^{-\xi_p^i (g^i_p - \rho^i)}} e^{-\xi_p^i g^i_p D},
\]
where
\[ g^i_p \triangleq \min_{n=1}^{n(p)} g^i_{(n)}, \beta^i_{(n)} = \frac{\beta(n) R(n)}{r^i}, \frac{1}{\xi_p^i} = \frac{1}{\alpha^i} + \sum_{n=1}^{n(p)} \frac{1}{\beta^i_{(n)}}, \text{ and } B_p \triangleq \sum_{n=1}^{n(p)} B_{(n)}. \]

**Proof.** See Section 2.6.7.

When packet size and propagation delays are not negligible, the end-to-end delay tail distribution is upper bounded as follows.

**Proposition 2.15** For every session \( i \) with a reserved rate \( r \leq r(p) \):

\[
\Pr \{ D(p) \geq D \} \leq \frac{\Lambda + \sum_{l \in p} B_l}{1 - e^{-\tilde{\beta}_l R_l}} e^{-\left( \frac{r - \sum_{l \in p} d_l/n(p) \cdot \bar{L}}{\tilde{\beta}_l + r \sum_{l \in p} \bar{d}_l} \right)}.
\]

where \( \tilde{\beta}_l = \beta_l \cdot R_l \)

**Proof.** The upper bound on the end-to-end delay tail distribution, given in (2.36), is derived from Proposition 2.14 by following the same steps as in Section 2.4.1.
Routing Algorithms

We seek a path from a source node $s$ to a destination $t$ that minimizes the end-to-end delay tail distribution upper bound in (2.36). Due to the complexity of the bound, we only provide approximated routing scheme under the following relaxing assumptions:

1. The coefficients $B_l$ at all links are equal, hence, $\sum_{l \in p} B_l = n(p) \cdot B$.

2. The coefficients $\bar{\beta}_l$ are from a relative small set $\bar{\beta}^1, \bar{\beta}^2, \ldots, \bar{\beta}^J$, and we assume $\bar{\beta}^1 \gg \bar{\beta}^2 \gg \ldots \gg \bar{\beta}^J$.

The second assumption implies that the sum $\sum_{l \in p} \frac{1}{\bar{\beta}_l}$ is dominated by the links of the smallest $\bar{\beta}_l$. We refer to these links as dominant links. Let $m_p(j)$ be the number of links, along $p$, from class $j$, and let $j_p^{\text{max}}$ be the dominant class (i.e., the class of the dominant links along $p$). Hence, the following approximation holds:

$$\sum_{l \in p} \frac{1}{\bar{\beta}_l} \approx \frac{m_p(j_p^{\text{max}})}{\bar{\beta}_p^{\text{max}}}.$$  \hspace{1cm} (2.37)

The routing scheme in Figure 2.14 finds the shortest path with respect to the metric $\{d_l\}$ for each possible number of hops $n$, $1 \leq n \leq H$, for each possible number of dominant links, $1 \leq m_p(j_p^{\text{max}}) \leq H$, for each possible dominant class, $1 \leq j_p^{\text{max}} \leq J$, and for each maximal rate $R^k$, $1 \leq k \leq K$.

We use the following notations:

- $p(v, k, n, j, m)$ - the shortest path with respect to the metric $d_l$, from the source $s$ to a node $v$, with at most $n$ hops, with maximal link class $j$, with $m$ links from class $l$ and with all links support the rate $R^k$.

- $d[v, k, n, j, m]$ - the length of the path $p(v, k, n, j, m)$ with respect to the metric $d_l$.

- $\pi[v, k, n, j, m]$ - five values that represent the parent of node $v$ along $p(v, k, n, j, m)$, namely the rate of the path, the number of hops along the path, the class of the dominant links, and the number of the dominant links.

The algorithm’s complexity is $O(M \cdot H^2 \cdot K \cdot J)$.

Consider the special cases in which all the links in the network belong to the same class, i.e., have the same parameter $\bar{\beta}_l = \bar{\beta}$. Then,

$$\sum_{l \in p} \frac{1}{\bar{\beta}_l} = \frac{n(p)}{\bar{\beta}} = \frac{m_p(j_p^{\text{max}})}{\bar{\beta}_p^{\text{max}}}.$$  \hspace{1cm} (2.38)
1. for each vertex \( v \in V \) do
   (a) for each \( 0 \leq n \leq H, 1 \leq k \leq K, 1 \leq j \leq J, 0 \leq m \leq n \) do
      i. \( d[v, k, n, j, m] \leftarrow \infty \)
      ii. \( \pi[v, k, n, j, m] \leftarrow NIL \)

2. for each \( 1 \leq k \leq K, 1 \leq j \leq J \) do
   (a) \( d[s, k, 0, j, 0] \leftarrow 0 \)

3. for \( k \leftarrow 1 \) to \( K \) do
   (a) delete all links \( l \) with \( R_l^t < R^k \)
   (b) for \( j \leftarrow 1 \) to \( J \) do
      i. delete links \( l \) with \( \bar{\beta}_l < \bar{\beta}_j \)
      ii. for \( n \leftarrow 0 \) to \( H - 1 \) do
         A. for each edge \((u, v) \in E, u, v \in V\) do
            • if \( \bar{\beta}_{(u,v)} = \bar{\beta}_j \) then do
               for \( m \leftarrow 0 \) to \( n - 1 \) do
                  if \( d[v, k, n + 1, j, m + 1] > d[u, k, n, j, m] + d_{(u,v)} \) then do
                     \( d[v, k, n + 1, j, m + 1] \leftarrow d[u, k, n, j, m] + d_{(u,v)} \) and
                     \( \pi[v, k, n + 1, j, m + 1] \leftarrow \{u, k, n, j, m\} \)
            • else do
               for \( m \leftarrow 0 \) to \( n - 1 \) do
                  if \( d[v, k, n + 1, j, m] > d[u, k, n, j, m] + d_{(u,v)} \) then do
                     \( d[v, k, n + 1, j, m + 1] \leftarrow d[u, k, n, j, m] + d_{(u,v)} \) and
                     \( \pi[v, k, n + 1, j, m] \leftarrow \{u, k, n, j, m\} \)

4. for each \( 1 \leq n \leq H, 1 \leq k \leq K, 1 \leq j \leq J, 1 \leq m \leq H \) compute,

\[
\Pr(D, p(t, k, n, j, m)) = \frac{\Lambda + n \cdot B}{1 - \epsilon \left( \frac{R^k - \sum_{l \in p} d_l}{\pi + R^k \frac{m}{\beta^j}} \right)}
\]

5. among \( O(H^2 \cdot K \cdot J) \) paths choose a path \( \tilde{p} \) with the minimal \( \Pr(D, \tilde{p}) \)

Figure 2.14: Algorithm Minimum Delay tail Distribution - Approximation (MDD-A)
For this case, the following proposition holds.

**Proposition 2.16** The path \( \tilde{p} \) identified by Algorithm Minimum Delay tail Distribution - Approximation (MDD-A) minimizes the end-to-end delay tail distribution bound.

**Proof.** The algorithm identifies the shortest path with respect to the metric \( \{d_\ell\} \) for all possible maximal rates, for all possible number of hops, for all possible values of \( \sum_{\ell \in p} B_\ell \), and for all possible values of \( \sum_{\ell \in p} \bar{\beta}_\ell \). The proposition follows immediately. \( \blacksquare \)

In general, equation (2.38) does not (strictly) hold; yet, due to assumption (2.37), the algorithm identifies a near-optimal solution.

2.5 Conclusion

The study in this chapter contributes to two major subjects within the area of QoS provision. First, it considers QoS routing schemes that are designed to operate in conjunction with rate-based service disciplines. While previous studies (e.g., [8, 10, 9]) exclusively dealt with (deterministically) burstiness constrained (BC) traffic, deterministic guarantees and fixed-rate links, in this study we (mainly) investigated the problem within the realm of stochastically bounded burstiness (SBB) and stochastic guarantees; moreover, we also investigated the important case of variable-rate links.

The second contribution of this study was in fact a prerequisite for the first. Namely, in order to establish appropriate routing schemes, we needed to have at hand end-to-end delay bounds for networks of packetized servers and links with non-negligible propagation delays. Since previous work on stochastic (EBB, SBB) settings have been carried on more limited models (at times, on a single, isolated server), we had to make the required extensions. As a result, the present study is the first to provide end-to-end bounds in a “full” network model, for the settings of EBB and SBB traffic with both constant and variable rate links.

The new bounds have a much more complex structure than the deterministic bound of the “basic” BC setting. Moreover, the way they should be employed within a corresponding routing scheme is not as straightforward as in the basic setting. Yet, once the right observations are made, the complexity of the resulting QoS routing scheme is typically not higher than in the basic setting. Special care is often required also when attempting to adopt the rate quantization approach of [9], e.g., as demonstrated by the non-uniform quantization method applied in algorithm QP-RQ. After having established the routing schemes, we tested them by means of simulation examples. Through these examples, we demonstrated that (i) our MDD routing algorithm correctly identifies a “stochastically”-feasible path,
whereas a corresponding deterministic routing algorithm may identify a false path that is not “stochastically”-feasible, and (ii) our MC routing algorithm achieves better network utilization (in terms of blocking probability) than the standard shortest-widest and widest-shortest routing algorithms. It is important to note that, while our algorithms rely on somewhat complex analysis, their computational complexity is polynomial and quite comparable to that of simple shortest path algorithms, as those employed by standard routing protocols.

To conclude, the aggregate result of this chapter is the establishment of a framework for QoS provision and routing that is suitable for a wide range of applications and traffic characteristics, as well as for a wide range of network environments. Specifically, it allows to cope with packetized traffic of stochastically bounded burstiness and links with propagation delays and variable rates, and to better cope with applications that are satisfied with stochastic guarantees.

2.6 Appendix

2.6.1 Proof of Proposition 2.4

Proposition 2.4: For every session $i$ with BC traffic in an RPPS GPS network with FC service rate, we have:

$$Q^i_p (t) \leq \sigma^i + \frac{\rho^i}{r^i (p)} \cdot \sum_{n=1}^{n(p)} \frac{r^i(n)}{R(n)} \cdot \Delta(n), \quad (2.39)$$

$$D^i_p (t) \leq \sigma^i + \frac{1}{r^i (p)} \cdot \sum_{n=1}^{n(p)} \frac{r^i(n)}{R(n)} \cdot \Delta(n). \quad (2.40)$$

Proof. According to Lemma 4.1 in [38]:

$$\sum_{n=1}^{n(p)} Q^i_{n(p)} (t) = \sup_{0 \leq \tau \leq t} \left\{ A^i (\tau, t) - \inf_{\tau = \tau_1 \leq \cdots \leq \tau_{n(p)+1} = t} \left\{ \sum_{n=1}^{n(p)} S^i_{n(n)} (\tau_n, \tau_{n+1}) \right\} \right\} \quad (2.41)$$

The infimum is attained by the set of $\tilde{\tau}_n$, $1 \leq n \leq n (p) + 1$ such that $[\tilde{\tau}_n, \tilde{\tau}_{n+1})$ is contained in the session $i$ busy period at hop $n$ starting from $\tilde{\tau}_n$. For this set of $\tilde{\tau}_n$'s, $1 \leq n \leq n (p) + 1$, we have

$$\hat{S}^i_{n(p)} (\tau, t) \triangleq \inf_{\tau = \tau_1 \leq \cdots \leq \tau_{n(p)+1} = t} \left\{ \sum_{n=1}^{n(p)} S^i_{n(n)} (\tau_n, \tau_{n+1}) \right\} = \sum_{n=1}^{n(p)} \hat{S}^i_{n(n)} (\tilde{\tau}_n, \tilde{\tau}_{n+1})$$

Since the $n$-th hop is busy at the period $[\tilde{\tau}_n, \tilde{\tau}_{n+1})$,

$$S^i_{n(n)} (\tilde{\tau}_n, \tilde{\tau}_{n+1}) \geq \left[ r^i_{(n)} (\tilde{\tau}_{n+1} - \tilde{\tau}_n) - \frac{r^i_{(n)}}{R(n)} \cdot \Delta(n) \right]^+.$$
Hence,
\[
\hat{S}_i^{(n(p))} (\tau, t) = \sum_{n=1}^{n(p)} S_i^{(n)} (\tau_n, \tau_{n+1}) \geq \left[ r_i^p (t - \tau) - \sum_{n=1}^{n(p)} \frac{r_i^{(n)}}{R_i^{(n)}} \cdot \Delta (n) \right]^+ ,
\]
where \( r_i^p \triangleq \min_{n \in p(i)} r_i^{(n)} \).

Substituting the above into (2.41), we have
\[
Q_i^p (t) \leq \sup_{0 \leq \tau \leq t} \left\{ A_i^p (\tau, t) - \left[ r_i^p (t - \tau) - \sum_{n=1}^{n(p)} \frac{r_i^{(n)}}{R_i^{(n)}} \cdot \Delta (n) \right]^+ \right\} \tag{2.42}
\]
(2.42) implies that \( \hat{S}_i^{(n(p))} (\tau, t) \) is FC, and the total backlog in the network equals to the backlog in a single FC server. (2.39) and (2.40) follow from Proposition 2.3.

2.6.2 Proof of Proposition 2.6

**Proposition 2.6:** For every session \( i \) with EBB traffic in an RPPS PGPS network, and for any \( D > 0 \):
\[
\Pr \left\{ \tilde{D}_i^p (t) \geq D \right\} \leq \Lambda_i^p e^{-\alpha_i (D - \sum_{n=1}^{n(p)} \frac{\Delta (n)}{R_i^{(n)}})^+ n(p) \cdot L_i} . \tag{2.43}
\]

**Proof.** We base the proof of Proposition 2.6 on the decomposition approach introduced in [38]. Accordingly, each RPPS PGPS server \( l \) of rate \( R_l \) and \( I \) input sessions, is decomposed into a set of \( I_l \) virtual servers with rates \( g_{l1}^i, g_{l2}^i, \ldots, g_{lI_l}^i \), so that instead of having a GPS system with \( I_l \) sessions sharing a server, we have a decomposed system consisting of \( I_l \) separate queues, each of which has a dedicated server. For each session \( i \), let \( \delta_i^l (t) \) denote its backlog at time \( t \) served by a dedicated server of rate \( g_{li}^i \), in the virtual decomposed system. For each server \( l \) along the path \( p \) of session \( i \) we have: \( Q_i^l (t) \leq \delta_i^l (t) \), since under our RPPS GPS assignment a backlog clearing rate of \( g_{li}^i = \frac{r_i^p}{\sum_{j=1}^{r_i^p} R_j} \cdot R_l \) is guaranteed. From the study of G/G/1 queue, it is known that,
\[
\delta_i^l (t) = \sup_{0 \leq s \leq t} \left\{ A_i^l (s, t) - g_{li}^i (t - s) \right\} .
\]
Define,
\[
\delta_i^l (t) = \sup_{0 \leq s \leq t} \left\{ A_i^l (s, t) - g_{li}^i \cdot (t - s) \right\} .
\]
For any \(1 \leq n \leq n(p)\) and any time interval \([\tau, t]\), we have:
\[
Q^i_{(n)}(t) = Q^i_{(n)}(\tau) + A^i_{(n)}(\tau, t) - S^i_{(n)}(\tau, t),
\]

hence, for \(p = 1, \ldots, n(p)\),
\[
\sum_{n=1}^{p} Q^i_{(n)}(t) - \sum_{n=1}^{p} Q^i_{(n)}(\tau) = \sum_{n=1}^{p} [A^i_{(n)}(\tau, t) - S^i_{(n)}(\tau, t)].
\]

We assume that the system is empty at time \(t = 0\). Thus, \(\sum_{n=1}^{p} Q^i_{(n)}(0) = 0\).

With \(t = \tau_{n+1}\) and \(\tau = \tau_{n}\), \(1 \leq n \leq p\),
\[
\sum_{l=1}^{n} Q^i_{(l)}(\tau_{n+1}) - \sum_{l=1}^{n} Q^i_{(l)}(\tau_{n}) = \sum_{l=1}^{n} [A^i_{(l)}(\tau_{n}, \tau_{n+1}) - S^i_{(l)}(\tau_{n}, \tau_{n+1})].
\]

Summing up from \(k = 1\) to \(p\),
\[
\sum_{l=1}^{p} Q^i_{(l)}(\tau_{p+1}) - \sum_{n=1}^{p} Q^i_{(n)}(\tau_{n}) = \sum_{n=1}^{p} \left\{ \sum_{l=1}^{n} [A^i_{(l)}(\tau_{n}, \tau_{n+1}) - S^i_{(l)}(\tau_{n}, \tau_{n+1})] \right\},
\]

therefore,
\[
\sum_{l=1}^{p} Q^i_{(l)}(\tau_{p+1}) - \sum_{n=1}^{p} Q^i_{(n)}(\tau_{n}) = \sum_{n=1}^{p} [A^i_{(n)}(\tau_{n}, \tau_{n+1}) - S^i_{(n)}(\tau_{n}, \tau_{n+1})]
\]  
\[+ \sum_{n=2}^{p} [A^i_{(n)}(\tau_{n}, \tau_{p+1}) - S^i_{(n-1)}(\tau_{n}, \tau_{p+1})].\]  
(2.44)

Since \(\sum_{n=1}^{p} Q^i_{(n)}(\tau_{n}) \geq 0\), the following inequality holds,
\[
\sum_{n=1}^{p} Q^i_{(n)}(\tau_{p+1}) \geq \sum_{n=1}^{p} [A^i_{(n)}(\tau_{n}, \tau_{n+1}) - S^i_{(n)}(\tau_{n}, \tau_{n+1})]
\]  
\[+ \sum_{n=2}^{p} [A^i_{(n)}(\tau_{n}, \tau_{p+1}) - S^i_{(n-1)}(\tau_{n}, \tau_{p+1})].\]  
(2.45)

The equality holds if and only if \(\sum_{n=1}^{p} Q^i_{(n)}(\tau_{n}) = 0\).

Let \(\tau_{p+1} = t\).
From (2.45) and (2.44), we have

$$
\sum_{n=1}^{p} Q_{(n)}^i(t) = \sup_{0 \leq \tau_1 \leq \cdots \leq \tau_{p+1} \leq t} \left\{ \sum_{n=1}^{p} \left[ A_{(n)}^i(\tau_n, \tau_{n+1}) - S_{(n)}^i(\tau_n, \tau_{n+1}) \right] 
+ \sum_{n=2}^{p} \left[ A_{(n)}^i(\tau_n, \tau_{p+1}) - S_{(n-1)}^i(\tau_n, \tau_{p+1}) \right] \right\}. \quad (2.46)
$$

Since the service is non-cut-through, we have

$$
A_{(n)}^i(\tau, t) \leq S_{(n-1)}^i(\tau, t) + L^i.
$$
Hence,

$$
\sum_{n=2}^{p} \left[ A_{(n)}^i(\tau_n, \tau_{p+1}) - S_{(n-1)}^i(\tau_n, \tau_{p+1}) \right] \leq (p-1) L^i. \quad (2.47)
$$

From (2.46), (2.47), and the definition of $\hat{S}_{(p)}^i(\tau, t)$, we have

$$
\sum_{n=1}^{p} Q_{(n)}^i(t) \leq \sup_{0 \leq \tau \leq t} \left\{ A^i(\tau, t) - \hat{S}_{(p)}^i(\tau, t) \right\} + (p-1) L^i. \quad (2.48)
$$

It can be shown [38] that

$$
\hat{S}_{(p)}^i(\tau, t) \geq g_p^i \cdot (t - \tau),
$$
thus,

$$
\sum_{n=1}^{p} Q_{(n)}^i(t) \leq \sup_{0 \leq \tau \leq t} \left\{ A^i(\tau, t) - g_p^i \cdot (t - \tau) \right\} + (p-1) L^i. \quad (2.48)
$$

Consider the total backlog in the network, (2.48) implies that

$$
Q_{(p)}^i(t) - n(p) L^i \leq \delta^i(t), \quad (2.49)
$$
i.e. the total backlog in the network equals to the backlog in a single server with rate $r^i(p)$. Note, that equation (2.49) holds for any arrival process. We now incorporate the EBB characteristic of the arrival process. According to Theorem 1 in [13] the backlog is Exponentially Bounded (EB), thus

$$
\Pr\left\{ Q_{(p)}^i(t) \geq Q \right\} \leq \Lambda_p e^{-\alpha^i (Q - n(p) L^i)},
$$
and

$$
\Pr\left\{ D_{(p)}^i(t) \geq D \right\} \leq \Lambda_p e^{-\alpha^i (g_p^i \cdot D - n(p) L^i)}. \quad (2.50)
$$

Since packets are served non-preemptively, we have

$$
\Pr\left\{ \tilde{D}_{(p)}^i(t) \geq D \right\} \leq \Lambda_p e^{-\alpha^i \left( g_p^i \left( D - \sum_{n=1}^{n(p)} \frac{L}{H(n)} \right) - n(p) L^i \right)}.
$$
2.6.3 Proof of Proposition 2.8

**Proposition 2.8:** Algorithm MC correctly identifies the $q$-feasibility of the connection. Whenever the connection is $q$-feasible, the path $p(\tilde{n}, \tilde{k})$ and rate $r(\tilde{n}, \tilde{k})$, identified by the algorithm, achieve the minimal cost among all $q$-feasible solutions. The algorithm’s complexity is $O(M \cdot H \cdot K)$.

**Proof.** The first part of the proposition is straightforward. Consider, then, a $q$-feasible connection. Let $p^*$ and $r^*$ be, correspondingly, the path and rate that minimize the cost $MC$ among all $q$-feasible solutions. We thus have:

$$\bar{D}(p^*, r^*, q) \leq \bar{D},$$

where,

$$\bar{D}(p, r, q) = \begin{cases} 
\sum_{l \in p} d_l + \frac{n(p)k}{r} - \frac{1}{\alpha r} \ln \left( \frac{q}{\Lambda} \right) + \frac{1}{\alpha r} \ln \left( \frac{r}{r-\rho} \right) + \frac{\rho}{\alpha r(r-\rho)} \ln \left( \frac{\lambda}{\rho} \right) & r \leq \rho (\Lambda + 1) \\
\sum_{l \in p} d_l + \frac{n(p)L}{r} - \frac{1}{\alpha r} \ln (q) + \frac{\rho}{\alpha r(r-\rho)} \ln (\Lambda + 1) & r > \rho (\Lambda + 1)
\end{cases}$$

Let $k^*$ be the index for which $R^{k^*} = r(p^*)$. Clearly,

$$R^{k^*} \geq r^*.$$  \hspace{1cm} (2.51)

Consider the path $p(n(p^*), k^*)$, identified at the $k^*$-th iteration of the algorithm, for the value of $n(p^*)$ hops. Denote $\hat{p} = p(n(p^*), k^*)$. Clearly, $r(\hat{p}) \geq R^{k^*}$, therefore, by (2.51),

$$r^* \leq r(\hat{p}),$$ \hspace{1cm} (2.52)

i.e., the rate $\hat{r}$ can be supplied to the connection over path $\hat{p}$.

Clearly, by the algorithm,

$$\sum_{l \in \hat{p}} d_l \leq \sum_{l \in p^*} d_l$$ \hspace{1cm} (2.53)

and

$$n(\hat{p}) \leq n(p^*).$$ \hspace{1cm} (2.54)

Note that, $f(r) = \ln \left( \frac{r}{r-\rho} \right) + \frac{\rho}{r-\rho} \ln \left( \frac{\lambda}{\rho} \right)$ is a non-decreasing function for all $r > \rho$ (see the proof of proposition 2.7). (2.51)-(2.54) imply that

$$\bar{D}(\hat{p}, r^*, q) \leq \bar{D}(\hat{p}, r^*, q) \leq \bar{D}(\hat{p}, r^*, q) \leq D,$$ \hspace{1cm} (2.55)

and since the cost $C(r, n(p), \bar{D}(p, r, q))$ is non-decreasing in each variable,

$$C(r^*, \hat{p}) = C(r^*, n(\hat{p}), \bar{D}(\hat{p}, r^*, q)) \leq C(r^*, n(p^*), \bar{D}(p^*, r^*, q)) = C(r^*, p^*),$$ \hspace{1cm} (2.56)
which implies the second part of the proposition.

The third part of the proposition is straightforward. □

2.6.4 Proof of Proposition 2.9

Proposition 2.9: The effective end-to-end delay of the path \( p(\tilde{j}) \), identified by algorithm QP-RQ, is at most \( 1 + \epsilon \) times larger than the minimal value, i.e., if \( p^* \) is the optimal path then

\[
\bar{D}(p(\tilde{j}), q) \leq (1 + \epsilon) \cdot \bar{D}(p^*, q).
\]

The algorithm’s complexity is

\[
O \left( (N \log N + M) \cdot \min \left\{ \frac{1}{\epsilon} \log \frac{\rho \Lambda(R^K - \rho)}{(R^1 - \rho)}, K \right\} \right).
\]

Proof. We first show that for any \( a_{j+1} > 1 \) and \( R(j) > \rho \),

\[
\frac{1}{a_{j+1} \cdot R(j)} \ln \left( \frac{R(j)}{a_{j+1} \cdot R(j) - \rho} \right) + \frac{\rho}{a_{j+1} \cdot R(j) - \rho} \ln \left( \frac{R(j)}{\rho} \right) \leq \frac{a_{j+1} \cdot R(j) - \rho}{R(j) - \rho}.
\]

Consider the function \( f(x) = x \ln x - (x - \rho) \ln (x - \rho) \), it is easy to see that it is a non-decreasing function for all \( x > \rho \), thus,

\[
f(x) \leq f(a \cdot x) \quad \forall x > \rho, a > 1.
\]

With \( x = R(j) \) and \( a = a_{j+1} \), we have

\[
R(j) \ln (R(j)) - (R(j) - \rho) \ln (R(j) - \rho) \\
\leq (a_{j+1} \cdot R(j)) \ln (a_{j+1} \cdot R(j)) - (a_{j+1} \cdot R(j) - \rho) \ln (a_{j+1} \cdot R(j) - \rho).
\]

Applying some mathematic manipulations, we have

\[
(R(j) - \rho) \ln \left( \frac{R(j)}{R(j) - \rho} \right) + \frac{\rho (R(j) - \rho)}{(R(j) - \rho)} \ln \left( \frac{R(j)}{\rho} \right) \\
\leq (a_{j+1} \cdot R(j) - \rho) \ln \left( a_{j+1} \cdot R(j) - \rho \right) + \rho (a_{j+1} \cdot R(j) - \rho) \ln \left( \frac{a_{j+1} \cdot R(j)}{\rho} \right).
\]

From (2.59) it is clear that (2.57) holds.
Assume that the available rate at the optimal path $\mathbf{p}^*$ is in rate class $j^*$, i.e. $R(j^*) \leq r(\mathbf{p}^*) < a_{j^*+1} \cdot R(j^*)$, and let $\mathbf{p}(j^*)$ be the shortest (with respect to the metric $\left\{ \frac{L}{R(j^*)} + d_l \right\}$) path identified by the algorithm at its $j^*$-th iteration.

Consider, first, $r(\mathbf{p}^*) \leq \rho \cdot (\Lambda + 1)$.

Recall that

$$\bar{D}(\mathbf{p}, q) = \sum_{l \in \mathbf{p}} d_l + \frac{n(\mathbf{p})L}{r(\mathbf{p})} - \frac{1}{\alpha r(\mathbf{p})} \ln \left( \frac{q}{\Lambda} \right) + \frac{1}{\alpha r(\mathbf{p})} \ln \left( \frac{r(\mathbf{p})}{R(\mathbf{p})} \right) + \frac{\rho}{\alpha r(\mathbf{p})} \ln \left( \frac{r(\mathbf{p})}{\rho} \right)$$

(2.60)

Clearly, by the algorithm,

$$\sum_{l \in \mathbf{p}(j^*)} \left( d_l + \frac{L}{R(j^*)} \right) \leq \sum_{l \in \mathbf{p}^*} \left( d_l + \frac{L}{R(j^*)} \right).$$

(2.61)

(2.57), (2.60) and (2.61) imply that

$$\frac{\bar{D}(\mathbf{p}(j^*), q)}{\bar{D}(\mathbf{p}^*, q)} \leq \frac{a_{j^*+1} \cdot R(j^*) - \rho}{R(j^*) - \rho}.$$  

(2.62)

Now, we seek a quantization parameter $a_{j^*+1}$ for which

$$\frac{a_{j^*+1} \cdot R(j^*) - \rho}{R(j^*) - \rho} \leq (1 + \epsilon).$$

(2.63)

This parameter can be found by recursion, or alternatively, it can be easily verified that with

$$a_{j^*+1} \leftarrow \frac{\rho + (1 + \epsilon) \frac{(j^*+1)}{2} (R_1 - \rho)}{\rho + (1 + \epsilon) \frac{j^*}{2} (R_1 - \rho)},$$

(2.63) holds.

Consider, next, $\rho \cdot (\Lambda + 1) < r(\mathbf{p}^*) \leq R^K$.

Recall that

$$\bar{D}(\mathbf{p}, q) = \sum_{l \in \mathbf{p}} d_l + \frac{n(\mathbf{p})L}{r(\mathbf{p})} - \frac{1}{\alpha r(\mathbf{p})} \ln (q) + \frac{1}{\alpha (r(\mathbf{p}) - \rho)} \ln (\Lambda + 1).$$

Now, it can be shown that

$$\frac{\bar{D}(\mathbf{p}(j^*), q)}{\bar{D}(\mathbf{p}^*, q)} \leq \frac{a_{j^*+1} \cdot R(j^*) - \rho}{R(j^*) - \rho},$$

(2.64)

and it can be easily verified that, with $a_{j^*+1} \leftarrow \frac{1+(1+\epsilon)\frac{(j^*+1)}{2}}{1+(1+\epsilon)\frac{j^*}{2}} \Lambda$,

$$\frac{\bar{D}(\mathbf{p}(j^*), q)}{\bar{D}(\mathbf{p}^*, q)} \leq (1 + \epsilon)$$
The number of iterations is
\[
\begin{cases}
\frac{2\log_2 \left( \frac{2R}{R_1} \right)}{\log_2 (1+\epsilon)} & R^1 \leq R(j) < \rho (\Lambda + 1) \\
\frac{\log_2 \left( \frac{R^K - \rho}{\rho \Lambda} \right)}{\log_2 (1+\epsilon)} & \rho (\Lambda + 1) < R(j) \leq R^K,
\end{cases}
\]
and the algorithm’s complexity is straightforward.

2.6.5 Proof of Proposition 2.11

Proposition 2.11: For every session \( i \) with SBB traffic in an RPPS PGPS network, for any \( \xi > 0, Q > 0 \) and \( D > 0 \):

\[
\Pr \{ Q^i_p (t) \geq Q \} \leq f \left( (Q - n(\mathbf{p}) \cdot L^i - \rho^i \cdot \xi)^+ \right)
\]

\[+ \frac{1}{(r(\mathbf{p}) - \rho^i) \xi} \int_{Q-n(\mathbf{p}) \cdot L^i - \rho^i \cdot \xi}^{\infty} f(u) \, du, \quad (2.66)\]

\[
\Pr \{ D^i_p (t) \geq D \} \leq f \left( (\eta^i(\mathbf{p}) - \rho^i \cdot \xi)^+ \right) + \frac{1}{(r(\mathbf{p}) - \rho^i) \xi} \int_{\eta^i(\mathbf{p}) - \rho^i \cdot \xi}^{\infty} f(u) \, du, \quad (2.67)\]

where, \( \eta^i(\mathbf{p}) \triangleq r(\mathbf{p}) \left( D - \sum_{l \in \mathbf{p}} d_l \right) - n(\mathbf{p}) \cdot L^i \).

Proof. According to (2.49), the total backlog in a network with non-cut-through RPPS PGPS-servers is bounded as follows

\[
Q^i_p (t) - n(\mathbf{p}) \cdot L^i \leq \delta^i (t),
\]
where

\[
\delta^i (t) = \sup_{0 \leq s \leq t} \{ A^i(s, t) - r(\mathbf{p}) \cdot (t - s) \}.
\]

Applying the results of [14] for an isolated network element with SBB input traffic, we have

\[
\Pr \{ Q^i_p (t) \geq Q \} \leq f \left( (Q - n(\mathbf{p}) \cdot L^i - \rho^i \cdot \xi)^+ \right)
\]

\[+ \frac{1}{(r(\mathbf{p}) - \rho^i) \xi} \int_{Q-n(\mathbf{p}) \cdot L^i - \rho^i \cdot \xi}^{\infty} f(u) \, du.\]
Since packets are served non-preemptively and propagation delays are non-negligible, the end-to-end delay is stochastically bounded as follows

$$\Pr \left\{ D_p(t) \geq D \right\} \leq f \left( \left( (\eta^i(p)) - \rho^i \cdot \xi \right) \right) + \frac{1}{(r(p) - \rho)} \int_{\eta^i(p) - \rho \xi}^{\infty} f(u) \, du,$$

where $\eta^i(p) = r(p) \left( D - \sum_{l \in p} \left( d_l + \frac{L_{\text{max}}}{R_l} \right) \right) - n(p) \cdot L^i$.

The term $\sum_{l \in p} \frac{L_{\text{max}}}{R_l}$ can be neglected, or alternatively, the constant delay $\frac{L_{\text{max}}}{R_l}$ at each queue can be accumulated with the propagation delays.

### 2.6.6 Proof of Proposition 2.12

**Proposition 2.12:** The path $\tilde{p}$ identified by the algorithm MDD-RQ, constitutes an $\epsilon$-optimal solution, i.e.:

$$\bar{Pr}((1 + \epsilon) \cdot D, \tilde{p}) \leq \bar{Pr}(D, p^*)$$

The algorithm’s complexity is

$$O \left( (N \log N + M) \cdot \min \left\{ \frac{1}{\epsilon} \log \left( \frac{R^K - \rho}{R^1 - \rho} \right), K \right\} \right).$$

**Proof.** We first show that for any $\alpha > 0$, $a_{j+1} > 1$ and $R(j) > \rho$,

$$\ln \left( 1 + \frac{1}{\alpha (R(j) - \rho)} \right) \leq \left( a_{j+1} \cdot R(j) - \rho \right) \cdot \ln \left( 1 + \frac{1}{\alpha (a_{j+1} \cdot R(j) - \rho)} \right).$$

Consider the function $f(x) = (x - \rho) \ln \left( 1 + \frac{1}{\alpha (x - \rho)} \right)$, it is easy to see that it is a non-decreasing function for all $x > \rho$, thus,

$$f(x) \leq f(a \cdot x) \quad \forall x > \rho, a > 1.$$

With $x = R(j)$ and $a = a_{j+1}$, we have

$$(R(j) - \rho) \ln \left( 1 + \frac{1}{\alpha (R(j) - \rho)} \right) \leq (a_{j+1} \cdot R(j) - \rho) \ln \left( 1 + \frac{1}{\alpha (a_{j+1} \cdot R(j) - \rho)} \right),$$

and (2.68) holds.

Assume that the available rate at the optimal path $p^*$ is in rate class $j^*$, i.e. $R(j^*) \leq r(p^*) < a_{j^*+1} \cdot R(j^*)$, and let $p(j^*)$ be the shortest (with respect to the
metric \( \left\{ \frac{L}{R(j^*)} + d_l \right\} \) path identified by the algorithm at its \( j^* \)-th iteration.

Recall that
\[
\Pr(D, p) = \Lambda \cdot e^{-\alpha r \left( D - \sum_{l \in p} d_l \right) - n(p) \cdot L \cdot \rho \xi} \left( 1 + \frac{1}{\alpha (r - \rho) \xi} \right) \\
+ B \cdot e^{-\beta r \left( D - \sum_{l \in p} d_l \right) - n(p) \cdot L \cdot \rho \xi} \left( 1 + \frac{1}{\beta (r - \rho) \xi} \right). \tag{2.70}
\]

Clearly, by the algorithm,
\[
\sum_{l \in p(j^*)} \left( d_l + \frac{L}{R(j^*)} \right) \leq \sum_{l \in p^*} \left( d_l + \frac{L}{R(j^*)} \right). \tag{2.71}
\]

Assume that \( \Pr(D, p^*) < 1 \), otherwise the delay distribution upper bound is meaningless.

From (2.70),(2.71) and (2.68) it can be shown that
\[
\Pr \left( \frac{a_{j^*+1} \cdot R(j^*) - \rho}{R(j^*) - \rho} \cdot D, p(j^*) \right) \leq \Pr(D, p^*). \tag{2.72}
\]

Now, we look for a quantization parameter \( a_{j^*+1} \) for which
\[
\frac{a_{j^*+1} \cdot R(j^*) - \rho}{R(j^*) - \rho} \leq 1 + \epsilon \tag{2.73}
\]

This parameter can be found by recursion, or alternatively, it can be easily verified that with
\[
a_{j^*+1} \left( \begin{array}{c} \frac{a_{j^*+1} \cdot R(j^*) - \rho}{R(j^*) - \rho} \\ \frac{R^1 - \rho}{R^1 - \rho} \end{array} \right) = \rho + (1 + \epsilon) \frac{(j^*+1)^{\omega}}{\omega} (R^1 - \rho),
\]

(2.73) holds.

The required number of iterations is
\[
\frac{2}{\epsilon} \log \left( \frac{R^K - \rho}{R^1 - \rho} \right),
\]
and the algorithm’s complexity is straightforward. \( \blacksquare \)

### 2.6.7 Proof of Proposition 2.14

**Proposition 2.14:** For every session \( i \) with EBB traffic in an RPPS GPS network with EBF service rate, we have:

\[
\Pr \left\{ Q_p^i(t) \geq Q \right\} \leq \frac{\Lambda^i + B_p}{1 - e^{-\xi_p \left( \rho_b - \rho^i \right)}} e^{-\xi_p Q}.
\]
\[
\Pr \{ D_i^p (t) \geq D \} \leq \frac{\Lambda^i + B_p}{1 - e^{-\xi_i \phi^i} e^{-\xi_i \phi^i} D},
\]

where

\[
g^i_p \triangleq \min_{n=1}^{n(p)} g^i_n, \quad \beta^i_n = \frac{\beta_n R_n}{\tau^i}, \quad \frac{1}{\xi^i} = \frac{1}{\alpha^i} + \sum_{n=1}^{n(p)} \frac{1}{\beta^i_n}, \quad \text{and } B_p \triangleq \sum_{n=1}^{n(p)} B_n.
\]

Proof. According to Lemma 4.1 in [38]:

\[
\sum_{n=1}^{n(p)} Q^i_n(t) = \sup_{0 \leq \tau \leq t} \left\{ A^i(\tau, t) - \inf_{\tau = \tau_1 \leq \cdots \leq \tau_{n(p)}+1 = t} \left\{ \sum_{n=1}^{n(p)} S^i_n(\tau_n, \tau_{n+1}) \right\} \right\}.
\]

The infimum is attained by the set of \( \tilde{\tau}_n, 1 \leq n \leq n(p) + 1 \) such that \([\tilde{\tau}_n, \tilde{\tau}_{n+1})\) is

contained in the session \( \tau_i \) busy period at hop \( n \) starting from \( \tilde{\tau}_n \). For this set of \( \tilde{\tau}_n \)'s, \( 1 \leq n \leq n(p) + 1 \), we have

\[
\hat{S}^i_{(n(p))}(\tau, t) \triangleq \inf_{\tau = \tau_1 \leq \cdots \leq \tau_{n(p)}+1 = t} \left\{ \sum_{n=1}^{n(p)} S^i_n(\tilde{\tau}_n, \tilde{\tau}_{n+1}) \right\} = \sum_{n=1}^{n(p)} S^i_n(\tilde{\tau}_n, \tilde{\tau}_{n+1})
\]

Now we show that \( \hat{S}^i_{(n(p))}(\tau, t) \) has EBF. Consider the event

\[
\left\{ \hat{S}^i_{(n(p))}(\tau, t) \leq g^i_p \cdot (\tau - t) + \Delta^i \right\} = \left\{ \sum_{n=1}^{n(p)} S^i_n(\tilde{\tau}_n, \tilde{\tau}_{n+1}) \leq g^i_p \cdot (\tau - t) + \Delta^i \right\},
\]

where \( g^i_{(p)} \triangleq \min_{n \in P(i)} g^i_n \).

It is clear that

\[
\left\{ \hat{S}^i_{(n(p))}(\tau, t) \leq g^i_p \cdot (\tau - t) - \Delta^i \right\} \subset \bigcup_{n=1}^{n(p)} \{ S^i_n(\tilde{\tau}_n, \tilde{\tau}_{n+1}) \leq g^i_p \cdot (\tilde{\tau}_{n+1} - \tilde{\tau}_n) - p_n \cdot \Delta^i \},
\]

where \( \sum_{n=1}^{n(p)} p_n = 1 \). Applying the union bound, we have

\[
\Pr \left\{ \hat{S}^i_{(n(p))}(\tau, t) \leq g^i_p \cdot (\tau - t) - \Delta^i \right\} \\
\leq \Pr \left\{ \bigcup_{n=1}^{n(p)} \{ S^i_n(\tilde{\tau}_n, \tilde{\tau}_{n+1}) \leq g^i_p \cdot (\tilde{\tau}_{n+1} - \tilde{\tau}_n) - p_n \cdot \Delta^i \} \right\} \\
\leq \sum_{n=1}^{n(p)} \Pr \{ S^i_n(\tilde{\tau}_n, \tilde{\tau}_{n+1}) \leq g^i_p \cdot (\tilde{\tau}_{n+1} - \tilde{\tau}_n) - p_n \cdot \Delta^i \}. \tag{2.76}
\]
For each \( n \)-th hop we have

\[
\Pr \left\{ S_{i(n)}^i (\tilde{r}_n, \tilde{r}_{n+1}) \leq g_p^i \cdot (\tilde{r}_{n+1} - \tilde{r}_n) - p_n \Delta^i \right\} \leq B^{(n)} \cdot e^{-\beta^{(n)} p_n \cdot \Delta^i} \tag{2.77}
\]

Substituting (2.77) into (2.76), we have

\[
\Pr \left\{ \hat{S}_{i(n)}^i (\tau, t) \leq g_p^i \cdot (\tau - t) - \Delta^i \right\} \leq \sum_{n=1}^{n(p)} B^{(n)} \cdot e^{-\beta^{(n)} p_n \cdot \Delta^i},
\]

We take \( p_n \) such that \( \beta_1^i \cdot p_1 = \beta_2^i \cdot p_2 = \cdots = \beta_{n(p)}^i \cdot p_{n(p)} \) and \( \sum_{n=1}^{n(p)} p_n = 1 \), i.e.

\[
\frac{1}{\beta^{(n)}(p)} = \sum_{n=1}^{n(p)} \frac{1}{p_n}.
\]

Hence,

\[
\Pr \left\{ \hat{S}_{i(n(p))}^i (\tau, t) \leq g_p^i \cdot (\tau - t) - \Delta^i \right\} \leq \left( \sum_{n=1}^{n(p)} B^{(n)} \right) \cdot e^{-\frac{\Delta^i}{\sum_{n=1}^{n(p)} \frac{1}{p_n}}}. \tag{2.78}
\]

(2.78) implies that \( \hat{S}_{i(n(p))}^i (\tau, t) \) has EBF, and the total backlog in the network equals to the backlog in a single EBF server. (2.74) and (2.75) follow from Proposition 2.13.
Chapter 3

The EDF scheduling discipline - deterministic setting

3.1 Introduction

Under the Rate Controlled Earliest Deadline First (RC-EDF) [7] service discipline, the traffic of each connection is reshaped at every node to ensure that the traffic offered to the scheduler conforms with specific characteristics. In particular, typical regulators enforce, at each server inside the network, the same traffic parameter control as the one performed at the network access point. Reshaping makes the traffic at each node more predictable and, therefore, simplifies the task of guaranteeing performance to individual connections. Consequently, end-to-end delay bounds can be computed as the sum of the worst-case delay bounds at each server along the path [7].

The Earliest Deadline First (EDF) scheduling discipline has been proven to be an optimal scheduling discipline in the sense that if a set of sessions is schedulable under any scheduling discipline \((i.e.,\) if the session’s packets can be scheduled in such a way that all of their deadlines are met\), then the set is also schedulable under EDF [40]. Also, the Rate-Controlled EDF was proven to outperform GPS in providing end-to-end delay guarantees in a network [4]. Consequently, the EDF scheduling discipline has been widely investigated. In particular, the establishment of schedulability conditions and efficient admission control schemes have been considered in [27, 28]. Furthermore, the establishment of efficient end-to-end bounds based on per-node traffic shaping has been studied in [4, 23]. Nevertheless, the corresponding QoS routing and resource assignment problems have not been fully explored. Some simple routing schemes that aim at finding a feasible path have been proposed [11, 29]. However, those schemes do not aim at optimizing the deadline assignment nor the route selection, in terms of maximizing the ability to accommodate future calls. Consequently, the authors of [31] have proposed new resource division \((i.e.,\) deadline assignment) policies. However, the schemes proposed
in [31] consider routing and resource division independently. Furthermore, although it is known that reshaping the traffic with potentially different parameters at each node in the network might reduce the obtained end-to-end delay bound [4], none of the above routing schemes considers such traffic reshaping. In other words, all the proposed routing schemes consider the same traffic parameters along the path, and in particular the same traffic parameters as at the entrance to the network. Clearly, such an assumption results in loose end-to-end delay guarantees and lower network resource utilization (i.e., higher blocking probability). Consequently, under such settings, GPS networks might outperform EDF networks in terms of session blocking probability, as suggested in [41].

Focusing on burstiness constrained traffic (also known as single-leaky-bucket traffic), we consider the joint problem of optimizing the traffic reshaping parameters along a path (to obtain lower end-to-end delay bounds) and identifying the quickest path, i.e., the path with the minimum end-to-end delay bound. Then, we turn to consider the more complex problem of optimizing the route choice in terms of balancing the loads and accommodating multiple connections.

The rest of the chapter is structured as follows. In Section 3.2, we formulate the model. Next, in Section 3.3, we discuss the EDF schedulability conditions and some prerequisite results. In Section 3.4, we consider the problem of finding feasible paths and in particular quickest paths. More specifically, we establish new routing schemes that identify the quickest path while optimizing the traffic reshaping parameters. Then, in Section 3.5, we turn to consider the problem of optimizing the route choice as well as the deadline assignment. In Section 3.6, we illustrate the efficiency of our routing schemes through simulations. Finally, in Section 3.7, we conclude the chapter.

### 3.2 Model Formulation

Given is a network across which sessions need to be routed. The network is represented by a directed graph $G(V,E)$, in which nodes represent switches and arcs represent links. $V$ is the set of nodes and $E$ is the set of interconnecting links; let $|V| = N$ and $|E| = M$.

Each link $l \in E$ is characterized by (i) a service rate $R_l$ and (ii) a constant delay value $d_l$, related to the link’s speed, propagation delay and maximal transfer unit.

We assume that the Rate-Controlled Earliest Deadline First (RC-EDF) service discipline is employed in each link $l \in E$. Accordingly, traffic from a particular connection entering a switch passes through a traffic shaper before being delivered to the scheduler. The traffic shaper regulates traffic, so that the output of the shaper satisfies certain pre-specified traffic characteristics. We consider the BC (Burstiness
Constrained) traffic model. The traffic shaper reshapes the incoming traffic by delaying packets so that the output is BC, and then delivers them to the scheduler. The EDF scheduler associates a deadline \( t + \delta^i \) with each packet of a session \( i \) that arrives at time \( t \). The packets are served in the order of their assigned deadlines. For ease of presentation, we assume a preemptive EDF scheduler (or, alternatively, negligible packet sizes).

We assume a source (“explicit”) QoS routing framework, in which link state information is exchanged and maintained up-to-date among network nodes for path computation. Routing decisions are based on the image of the network at the source node.

A session \( i \) in the network is characterized by the following parameters:

- Source and destination nodes \( s^i \) and \( t^i \), correspondingly.
- A maximal traffic burst \( \sigma^i_0 \).
- A traffic upper rate \( \rho^i_0 \).
- A required end-to-end delay bound \( D^i \).

A session should be routed through some path \( p \) between the corresponding source and destination nodes. Let \( H \) be the maximal possible number of hops in a path. We denote by \( n(p) \) the number of hops (i.e., links) of a path \( p \). We shall also denote by \( D^i(p) \) the guaranteed end to end delay to session \( i \) along path \( p \).

Denote the set of sessions at link \( l \in E \) by \( I_l \) and the number of sessions at link \( l \) by \( I_l \). Let \( I_{\text{max}} \) be the maximal number of sessions at any link, that is \( I_{\text{max}} = \max_{l \in E} I_l \). Also, denote the residual rate of a link \( l \) by \( R'_l \), where \( R'_l = R_l - \sum_{i \in I_l} \rho^i_0 \). Finally, let \( R'_{\text{max}} \) be the maximal residual rate in the network, that is \( R'_{\text{max}} = \max_{l \in E} R'_l \).

Without loss of generality, all quantities are normalized, such that the minimum (i.e., basic resolution) unit is “1”.

### 3.3 EDF Schedulability Conditions

We consider flows that are \((\sigma, \rho)\) - burstiness constrained. Let \( I_l \) be a set of flows entering an EDF scheduler in link \( l \) with a service rate \( R_l \). Assume that each session \( i \) is characterized by a \((\sigma^i, \rho^i)\)-burstiness constrained flow and a maximum packet queueing delay of \( \delta^i_l \). Then, the set \( I_l \) is EDF-schedulable if and only if the following two conditions hold:

(i) the stability condition, \( \sum_{i \in I_l} \rho^i < R_l \),
and (ii) the schedulability condition,
\[ R_l t \geq \sum_{i \in I_l} 1 (t - \delta_i^l) (\sigma^i + \rho^i (t - \delta_i^l)), \forall t \geq 0, \]
where \( 1 (t) = \begin{cases} 0 & t < 0 \\ 1 & \text{otherwise} \end{cases} \).

Following [28], we define the link work availability function \( F_l : [0, \infty) \rightarrow [0, \infty) \) as
\[ F_l (t) = R_l t - \sum_{i \in I_l} 1 (t - \delta_i^l) (\sigma^i + \rho^i (t - \delta_i^l)). \] (3.2)

A typical instance of \( F_l (t) \) is depicted in Figure 3.1. \( F_l (t) \) specifies the worst case amount of work (in bits) available at time \( t \) at the EDF scheduler in link \( l \), while still guaranteeing session \( i \) a maximum packet delay of \( \delta_i^l \), for \( 1 \leq i \leq I_l \). Clearly, the schedulability condition (3.1) becomes \( F_l (t) \geq 0 \forall t \geq 0 \). Accordingly, the state of a link \( l \in E \) is given by \( F_l (t) \).

It is easy to see that \( F_l (t) \) linearly increases with discontinuities at times \((\delta_i^l)_{i \in I_l}\), at which the function decreases in the amount of \((\sigma^i)_{i \in I_l}\). Note that the local minima of the function \( F_l (t) \) are obtained at times \((\delta_i^l)_{i \in I_l} \). Then, the schedulability condition is equivalent to \( F_l (u) \geq 0 \forall u \in (\delta_i^l)_{i \in I_l} \cup \{0\} \) [28]. Let us assume, without loss of generality, that the flows in \( I_l = \{1, 2, \ldots, I_l\} \) are ordered by \((\delta_i^l)\): \( i < j \Rightarrow \delta_i^l \leq \delta_j^l \forall i, j \in I_l \). Let \( \delta_0^l = 0 \). Denote the local minima values by \( w_i^l \), where \( w_i^l = 0 \) and \( w_i^l = F_l (\delta_i^l), i = 1, 2, \ldots, I_l \). Also, denote the slopes of \( F_l (t) \) by \( r_i^l \), where \( r_i^l = R_l - \sum_{j \leq i} \rho^j, i = 0, 1, \ldots, I_l \). Then, the work availability function is fully specified by the set \((\delta_i^l, w_i^l, r_i^l)_{0 \leq i \leq I_l}\). Therefore, the set \((\delta_i^l, w_i^l, r_i^l)_{0 \leq i \leq I_l}\) constitutes the link parameters, which specify the state of the link \( l \in E \). Obviously, this set should be updated each time a connection is admitted to the scheduler or leaves it. An efficient update algorithm can be found in [28].

Consider a link \( l \) and a pending session \( m \) with traffic parameters \((\sigma_0^m, \rho_0^m)\). Then, the minimum queueing delay \( \delta_{m,0}^l \) that can be guaranteed to \( m \) can be found
by determining the leftmost position of $\sigma_0^m + \rho_0^m \cdot t$ such that it is below the graph of $F_l(t)$ for all $t \geq 0$, as in Figure 3.1. An efficient scheme that is given the link and traffic parameters and calculates $\delta_{l,0}^m$ can be found in [28].

### 3.4 Finding Feasible Paths

In the context of EDF routing, we seek both a path and a deadline allocation at each link along the chosen path. Accordingly, we define a path-deadline assignment as follows.

**Definition 3.1** A path-deadline assignment is a path $(p, \delta)$, where $\delta = \{\delta_l\}_{l \in p}$ is the set of deadline assignments at each link $l \in p$.

We begin with the basic problem of identifying a feasible path-deadline allocation. If several feasible paths exist, we seek a path with the minimal end-to-end delay. Consider first the standard case, where the traffic at each node is shaped with the parameters at the entrance to the network. In this case, the problem is formulated as follows.

**Quickest Feasible Path Problem (QFP):** Given are a network $G(V,E)$, with a service rate $R_l$, a propagation delay $d_l$, and a work availability function $F_l(t)$ for each $l \in E$. Also, given is a session $m$ with source $s^m$, destination $t^m$, upper rate $\rho_0^m$, burst $\sigma_0^m$ and an end-to-end delay requirement $D^m$. Find a feasible path-deadline assignment $(p^m, \delta^m)$, i.e., a path between $s^m$ and $t^m$ and a deadline allocation $\{\delta_l^m\}_{l \in p^m}$, such that:

1. $D^m(p^m) = \sum_{l \in p^m} (\delta_l^m + d_l) \leq D^m$,

2. for all $l \in p^m$:

   $$F_l(t) - 1 (t - \delta_l^m) (\sigma_0^m + \rho_0^m (t - \delta_l^m)) > 0 \ \forall t > 0.$$  

If there are several such paths, the one with the minimum end-to-end delay bound is selected, i.e., a path with the minimum $D^m(p^m)$.

Distributed routing schemes that solve this problem have been proposed in several studies (e.g., [29, 11, 28, 42]). Those routing schemes are guaranteed to find a feasible path, if one exists. Basically, the schemes employ the following algorithm, which is specified here for completeness.

**Algorithm QFP (sketch):** For each link $l \in E$, calculate the minimum queueing delay $\delta_{l,0}^m$ that link $l$ can guarantee to session $m$, while maintaining

$$F_l(t) - 1 (t - \delta_{l,0}^m) (\sigma_0^m + \rho_0^m (t - \delta_{l,0}^m)) > 0, \ \forall t > 0.$$
Then, find the shortest path with respect to the metric \( \{ \delta_{0}^{m} + d_{l} \} \). If the sum of the queueing and propagation delays along the identified path \( p \) is at most \( D^{m} \), i.e., \( D^{m}(p^{m}) \leq D^{m} \), then return \( p, \delta_{l}^{m} = \delta_{0}^{m} \); otherwise, there is no feasible path.

Next, consider the more complex problem of identifying a feasible path while reshaping the traffic at each node along the path. It is known that the selection of the traffic reshaping parameters influences the schedulable region under RC-EDF [4]. In other words, proper selection of the traffic reshaping to be performed at each node may result in a lower end-to-end delay bound. For simplicity, we focus on the restricted case of single-leaky-buckets. Accordingly, we define a path-reshaping-deadline assignment as follows.

**Definition 3.2** A path-reshaping-deadline assignment is a path \( (p, (\sigma, \rho), \delta) \), where \( (\sigma, \rho) = \{(\sigma_{l}, \rho_{l})\}_{l \in p} \) are the reshaping parameters at each link \( l \in p \) and \( \delta = \{\delta_{l}\}_{l \in p} \) is the deadline assignment at each link \( l \in p \).

Then, the problem is formulated as follows.

**Quickest Feasible Path with Traffic Reshaping Problem (QFPTS):**
Given are a network \( G(V, E) \), with a service rate \( R_{l} \), a propagation delay \( d_{l} \), and a work availability function \( F_{l}(t) \) for each \( l \in E \). Also, given is a session \( m \) with source \( s^{m} \), destination \( t^{m} \), upper rate \( \rho_{0}^{m} \), burst \( \sigma_{0}^{m} \) and an end-to-end delay requirement \( D^{m} \). Find a feasible path-reshaping-deadline assignment \( (p^{m}, (\sigma^{m}, \rho^{m}), \delta^{m}) \), i.e., a path between \( s^{m} \) and \( t^{m} \), reshaping parameters \( \{(\sigma_{l}^{m}, \rho_{l}^{m})\}_{l \in p^{m}} \), and a deadline allocation \( \{\delta_{l}^{m}\}_{l \in p^{m}} \), such that:

1. \( \max_{l \in p^{m}} \left( \frac{\sigma_{0}^{m} - \sigma_{l}^{m}}{\rho_{l}^{m}} \right) + \sum_{l \in p^{m}} (\delta_{l}^{m} + d_{l}) \leq D^{m} \)
2. for all \( l \in p^{m} \): \( \rho_{l}^{m} \geq \rho_{0}^{m} \),
3. for all \( l \in p^{m} \):
   \[
   F_{l}(t) - 1 (t - \delta_{l}^{m}) (\sigma_{l}^{m} + \rho_{l}^{m} (t - \delta_{l}^{m})) > 0, \quad \forall t > 0.
   \]

If there are several such paths, the one with the minimum end-to-end delay bound is selected, i.e., a path with the minimum

\[
D^{m}(p^{m}) = \max_{l \in p^{m}} \left( \frac{\sigma_{0}^{m} - \sigma_{l}^{m}}{\rho_{l}^{m}} \right) + \sum_{l \in p^{m}} (\delta_{l}^{m} + d_{l}).
\]

Note that the maximum reshaping delay \( i.e., \max_{l \in p^{m}} \left( \frac{\sigma_{0}^{m} - \sigma_{l}^{m}}{\rho_{l}^{m}} \right) \) is incurred only once, and is independent of the number of hops on the path [4].
3.4.1 Reshaping both the traffic burst as well as the traffic rate

Let $W_l(\sigma^m_l, \rho^m_l)$ be the minimum delay that can be guaranteed to a session $m$ at link $l$, as a function of the reshaping parameters $\sigma^m_l$ and $\rho^m_l$. Accordingly, for $0 \leq \sigma^m_l \leq \sigma^m_0$ and $\rho^m_0 \leq \rho^m_l < R^m_l$:

$$W_l(\sigma^m_l, \rho^m_l) = \{\min d | F_l(t) - 1(t - d) (\sigma^m_l + \rho^m_l (t - d)) > 0 \forall t > 0\}.$$  

Obviously, we have that $W_l(\sigma^m_0, \rho^m_0) = \delta^m_{l,0}$. Furthermore, given the session $m$ traffic reshaping parameters at each link $l \in \mathbf{p}$, $(\sigma^m_l, \rho^m_l)$, the end-to-end delay along a path $\mathbf{p}$ is given by

$$D^m(\mathbf{p}) = \max_{l \in \mathbf{p}} \left( \frac{(\sigma^m_0 - \sigma^m_l)^+}{\rho^m_l} \right) + \sum_{l \in \mathbf{p}} (W_l(\sigma^m_l, \rho^m_l) + d_l).$$

Suppose that the maximum allowed reshaping delay along a path $\mathbf{p}$ is $C$ time units, that is

$$\max_{l \in \mathbf{p}} \left( \frac{(\sigma^m_0 - \sigma^m_l)^+}{\rho^m_l} \right) \leq C.$$

Then, at each link $l$ we seek the session $m$ reshaping parameters $(\sigma^m_l, \rho^m_l)$ that minimize the link’s $l$ guaranteed delay, $W_l(\sigma^m_l, \rho^m_l)$. Denote the obtained minimum delay (with a maximum reshaping delay of $C$ time units) by $W_l(C)$. Clearly, $W_l(0) = \delta^m_{l,0}$. Algorithm Minimum-Link-Delay, specified in Figure 3.2, identifies the optimum reshaping parameters $(\bar{\sigma}^m_l, \bar{\rho}^m_l)$ and the corresponding minimum delay $W_l(C)$.

**Proposition 3.1** (a) Let $\mathcal{I}_l$ be a schedulable set of sessions at link $l$, and let $\{w^l_i, \delta^l_i, r^l_i\}_{0 \leq i \leq \mathcal{I}_l}$ be a set of parameters defining the link work availability function. Let $(\rho^m_0, \sigma^m_0)$ be the traffic parameters of a new session $m$, at the entrance to the network. Also, let $C$ be the maximal allowed reshaping delay. Then, the minimum link’s guaranteed delay $W_l(C)$, and the corresponding session $m$ reshaping parameters at link $l$, $(\bar{\sigma}^m_l, \bar{\rho}^m_l)$, are correctly identified by Algorithm Minimum-Link-Delay.

(b) The algorithm’s complexity is $O(I_l \cdot \log (\delta^m_{1,0}))$.

**Proof.** A maximal reshaping delay $C$ implies that, for all possible reshaping parameters $(\sigma^m_l, \rho^m_l)$, we have $\sigma^m_l + \rho^m_l C = \sigma^m_0$. Accordingly, a given delay $\lambda$ can be guaranteed only if $F_l(\lambda + C) > \sigma^m_0$. Keeping that in mind, the algorithm calculates the maximum rate $r_{\max}$ for which $F_l(t) - \sigma^m_0 - r_{\max} (t - \lambda - C) > 0$, $\forall t \geq \lambda + C$. Then, the algorithm calculates the minimum rate $r_{\min}$ for which $F_l(t) - (\sigma^m_0 - r_{\min} (\lambda + C - t)) > 0$ for all $t$ in the range $\lambda \leq t < \lambda + C$. Accordingly, the algorithm sets $\rho^m_l$ to be such that $\rho^m_l \geq \rho^m_0$ and $r_{\min} \leq \rho^m_l \leq r_{\max}$.  

68
input: \( \{w_i^l, \delta_i^l, r_i^l\}_{0 \leq i \leq I_l}, (\rho_0^m, \sigma_0^m), C \), output: \( (\bar{\sigma}_l^m, \bar{\rho}_l^m), W_l(C) \)

1. find \( \delta_{l,0}^m \)
2. \( H \leftarrow \delta_{l,0}^m \)
3. \( L \leftarrow 0 \)
4. \( \lambda \leftarrow \delta_{l,0}^m \)
5. while \( |\lambda - \frac{L+H}{2}| > 1 \)
   (a) \( \lambda \leftarrow \frac{L+H}{2} \)
   (b) let \( j \) such that \( \delta_{j-1}^l \leq \lambda + C \) and \( \delta_j^l > \lambda + C \)
   (c) let \( b \) such that \( \delta_{b-1}^l \leq \lambda \) and \( \delta_b^l > \lambda \)
   (d) if \( w_{j-1}^l + r_{j-1}^l (\lambda + C - \delta_{j-1}^l) > \sigma_0^m \)
      i. \( r_{max}^1 \leftarrow R_l^l \)
      ii. \( r_{max}^2 \leftarrow \min_{i \geq j} \frac{w_i^l - \sigma_0^m}{\delta_i^l - \lambda - C} \)
      iii. \( r_{max}^1 \leftarrow \min \{r_{max}^1, r_{max}^2\} \)
      iv. \( r_{min}^1 \leftarrow \frac{\sigma_0^m - (w_{j-1}^l + r_{j-1}^l (\lambda - \delta_{j-1}^l))}{C} \)
      v. \( r_{min}^2 \leftarrow \max_{b \leq i \leq j-1} \frac{\sigma_0^m - w_i^l}{\lambda + C - \delta_i^l} \)
      vi. \( r_{min} \leftarrow \max \{r_{min}^1, r_{min}^2\} \)
      vii. if \( \rho_0^m > r_{max} \) or \( r_{min} > r_{max} \) then go to step 5e
      viii. (else) if \( \rho_0^m > r_{min} \) then \( \rho_{l}^m \leftarrow \rho_0^m \)
      ix. else \( \rho_{l}^m \leftarrow r_{min} \)
      x. \( \sigma_{l}^m \leftarrow \sigma_0^m - \rho_{l}^m C \)
      xi. \( H \leftarrow \lambda \)
   (e) else \( L \leftarrow \lambda \) (a delay \( \lambda \) cannot be guaranteed)
6. \( W_l(C) \leftarrow \lambda \)
7. \( (\bar{\sigma}_l^m, \bar{\rho}_l^m) \leftarrow (\sigma_{l}^m, \rho_{l}^m) \)

Figure 3.2: Algorithm Minimum-Link-Delay
The algorithm considers $O \left( \log \left( \delta_{l,0}^m \right) \right)$ delay values, and for each delay $\lambda$ the algorithm involves $O \left( I_l \right)$ calculations. Thus the algorithm’s complexity is $O \left( I_l \cdot \log \left( \delta_{l,0}^m \right) \right)$.

1. for all possible values of $C$, $0 \leq C \leq \frac{\sigma_m^0}{\rho_0}$
   
   (a) for each $l \in E$, calculate $W_l \left( C \right)$ and $(\bar{\sigma}_l^m, \bar{\rho}_l^m)$ through algorithm Minimum-Link-Delay
   
   (b) find the shortest path $p \left( C \right)$ w.r.t. $\{W_l \left( C \right) + d_l\}$ (through Dijkstra’s shortest path algorithm)

2. let $\tilde{C}$ such that $\tilde{C} + \sum_{l \in p(\tilde{C})} \left( W_l \left( \tilde{C} \right) + d_l \right)$ is minimized

3. if $\tilde{C} + \sum_{l \in p(\tilde{C})} \left( W_l \left( \tilde{C} \right) + d_l \right) < D^m$
   
   (a) return $\tilde{p}^m \leftarrow p \left( \tilde{C} \right)$, $\left( \bar{\sigma}_l^m, \bar{\rho}_l^m \right)$, $\delta_l^m \leftarrow W_l \left( \tilde{C} \right)$
   
   (b) end

4. else there is no feasible path, end

Figure 3.3: Algorithm QFPTS

With the above algorithm at hand, we turn to solve the routing and traffic re-shaping problem QFPTS. Algorithm QFPTS, specified in Figure 3.3, solves this problem. The algorithm executes Dijkstra’s shortest path algorithm for all possible values of $C$. Thus, obviously, the complexity of the algorithm could be prohibitively large. Therefore, in order to reach an efficient yet computationally tractable solution, we establish an approximation scheme based on quantizing the reshaping delays. For ease of presentation, we begin with an approximation of order 2, i.e., the following algorithm shall derive a solution (path-reshaping-deadline assignment) whose guaranteed end-to-end delay is at most twice greater than that of the quickest.

The session $m$ reshaping delay $C$ could take any value in the interval $\left[ 0, \frac{\sigma_m^0}{\rho_0} \right]$. We group these values into $O \left( \log \frac{\sigma_m^0}{\rho_0} \right)$ reshaping-delay-classes, such that, for $1 \leq k \leq \lceil \log_2 \frac{\sigma_m^0}{\rho_0} \rceil$, reshaping-delay-class $k$ contains all reshaping delays in the range $\left( 2^{k-1}, 2^k \right]$. The Quantized 2-Approximation algorithm, termed Algorithm QFPTS-Q-2, is specified in Figure 3.4.
1. for all $l \in E$, calculate $\delta_{l,0}^m$

2. $C^0 \leftarrow 0$, $(\bar{\sigma}_l^m, \bar{\rho}_l^m)(0) \leftarrow (\sigma_{l,0}^m, \rho_{l,0}^m)$

3. find the shortest path $p(0)$ w.r.t. $D_{l,0}^m + d_l$

4. for all $k$, $1 \leq k \leq \lceil \log_2 \frac{\sigma_{l,0}^m}{\rho_{l,0}^m} \rceil$
   (a) $C^k \leftarrow 2^k$
   (b) for all $l \in E$, calculate $W_l(C^k)$ and $(\bar{\sigma}_l^m, \bar{\rho}_l^m)(k)$ through algorithm Minimum-Link-Delay
   (c) find the shortest path $p(k)$ w.r.t. $\{W_l(C^k) + d_l\}$ (through Dijkstra’s shortest path algorithm)

5. let $\tilde{k}$ such that $C^\tilde{k} + \sum_{l \in p(\tilde{k})} (W_l(C^\tilde{k}) + d_l)$ is minimized

6. if $C^\tilde{k} + \sum_{l \in p(\tilde{k})} (W_l(C^\tilde{k}) + d_l) < D^m$
   (a) return $\tilde{p}^m \leftarrow p(\tilde{k})$, $(\bar{\sigma}_l^m, \bar{\rho}_l^m) \leftarrow (\bar{\sigma}_l^m, \bar{\rho}_l^m)(\tilde{k})$, $\delta_l^m \leftarrow W_l(C^\tilde{k})$
   (b) end

7. else there is no feasible path, end

Figure 3.4: Algorithm QFPTS-Q-2

**Proposition 3.2** (a) The guaranteed end-to-end delay of the output path $\tilde{p}^m$ of algorithm QFPTS-Q-2 is at most twice larger than the minimal value, i.e., if $p^*$ is the quickest path then $D^m(\tilde{p}^m) \leq 2 \cdot D^m(p^*)$.

(b) The algorithm’s complexity is

$$O \left( MI_{\text{max}} \log(D) + \log \left( \frac{\sigma_{l,0}^m}{\rho_{l,0}^m} \right) (N \log N + M) \right).$$

**Proof.** Let $C^*$ be the reshaping delay of the optimal solution and $k^*$ be the corresponding reshaping-delay-class. Then, by the algorithm, $C^{k^*} \leq 2 \cdot C^*$. Furthermore, since $C^* \leq C^{k^*}$ implies that $W_l(C^{k^*}) \leq W_l(C^*)$, it holds that $\sum_{l \in p(k^*)} W_l(C^{k^*}) + d_l \leq \sum_{l \in p^*} W_l(C^*) + d_l$. Therefore,

$$C^{k^*} + \sum_{l \in p(k^*)} W_l(C^{k^*}) + d_l \leq 2 \cdot C^* + \sum_{l \in p^*} W_l(C^*) + d_l.$$
Then, clearly, \( D^m(\tilde{p}^m) \leq C^{k^*} + \sum_{l \in p^*} W_l(C^{k^*}) + d_l \), which implies that \( D^m(\tilde{p}^m) \leq 2 \cdot D^m(p^*) \), and the first part of the proposition follows. The second part of the proposition is straightforward.

We proceed to generalize the above algorithm for obtaining an \( \epsilon \)-approximation. Given a value \( \epsilon > 0 \), the reshaping delay values are grouped into \( O(\log \frac{\sigma_m}{\rho_0}) \) reshaping-delay-classes, such that, for \( 1 \leq k \leq [\log_{1+\epsilon}(\frac{\sigma_m}{\rho_0})] \), reshaping-delay-class \( k \) contains all reshaping delays in the range \((1 + \epsilon)^{k-1}, (1 + \epsilon)^k\). The corresponding Algorithm QFPTS-Q-\( \epsilon \) is identical to Algorithm QFPTS-Q-2, except that it consists of \( O(\log_{1+\epsilon}(\frac{\sigma_m}{\rho_0})) \) iterations that consider the above \( O(\log_{1+\epsilon}(\frac{\sigma_m}{\rho_0})) \) reshaping-delay-classes.

**Proposition 3.3** (a) The guaranteed end-to-end delay of the output path \( \tilde{p}^m \) of Algorithm QFPTS-Q-\( \epsilon \) is at most \( 1 + \epsilon \) larger than the minimum value, i.e., if \( p^* \) is a quickest path, then \( D^m(\tilde{p}^m) \leq (1 + \epsilon) \cdot D^m(p^*) \).
(b) The algorithm’s complexity is \( O(M_{max} \log (D) + \frac{1}{\epsilon} \log (\frac{\sigma_m}{\rho_0}) (N \log N + M)) \).

**Proof.** The proof of the first part of the proposition is similar to that of Proposition 3.2. The second part is immediate from the following relation:

\[
O \left( \log_{1+\epsilon}(\frac{\sigma_m}{\rho_0}) \right) = O \left( \frac{\log(\frac{\sigma_m}{\rho_0})}{\log (1 + \epsilon)} \right) = O \left( \frac{1}{\epsilon} \cdot \log(\frac{\sigma_m}{\rho_0}) \right).
\]

We now explain how the above approximation scheme can admit a distributed implementation. The scheme executes a shortest path algorithm \( O \left( \frac{1}{\epsilon} \log(\frac{\sigma_m}{\rho_0}) \right) \) times, each time for a different value of a maximum reshaping delay \( C \). Accordingly, a distributed version would employ the distributed Bellman-Ford shortest path algorithm \( O \left( \frac{1}{\epsilon} \log(\frac{\sigma_m}{\rho_0}) \right) \) times, for predefined values of \( C \) (or, alternatively, for values that are flooded by the source node). Among all identified paths, the source node would choose the optimal one. As for Algorithm Minimum-Link-Delay, it is easy to verify that it can be executed at each link distributively. We note that, through similar lines, all the routing schemes presented here can be adapted into distributed versions.

**An Example**
We now demonstrate the efficiency of the proposed QFPTS algorithm. Through a simple network example, we indicate that the standard QFP algorithm may not
identify a feasible path; in contrast, our QFPTS algorithm is guaranteed to identify a feasible path, if it exists.

Consider some source and destination nodes connected by a 6-hop path, as depicted in Figure 3.5. The bandwidth of each of the six links is assumed to be 10Mbps. Assume that propagation delays are negligible. Suppose that a session, referred to as session 1, with (1MB, 2Mbps)-burstiness constrained traffic, is routed through these links. Also, assume that session 1 is reshaped at each link with the same traffic parameters as at the entrance to the network. Furthermore, assume that the session is assigned a deadline of 200ms at each link, which constitute an end-to-end guaranteed delay of 1200ms. Now, suppose that a new session, referred to as session 2, is pending, and suppose that its traffic parameters are (1MB, 5Mbps), and it requires an end-to-end delay guaranteed of 1000ms. Under the above setting, the standard QFP algorithm assumes a minimum delay of 200ms at each link. A minimum delay of 200ms follows since at time $t = 200ms$ the work availability function equals to 1MB, that is, $F_l(200ms) = 10Mbps \cdot 200ms - 1MB = 1MB$. Accordingly, the minimum end-to-end delay along the path is 1200ms, which is not feasible. In contrast, our QFPTS algorithm identifies an optimal reshaping at each node; that is, the traffic is reshaped with the parameters (0MB, 5Mbps), which corresponds to a shaping delay of 500ms. Consequently, a minimum queueing delay of 0ms can be guaranteed at each link. A 0ms delay follows since $F_l(t) - 5Mbps \cdot t \geq 0$, $\forall t > 0$.

In particular, at time $t = 200ms$, $F_l(t) - 5Mbps \cdot t = 1MB - 5Mbps \cdot 200ms = 0$. Thus, the path is feasible for session 2 with a guaranteed end-to-end delay of 500ms. Alternatively, algorithm QFPTS-Q-2 identifies a guaranteed end-to-end delay of at most 1000ms, which is still feasible.

3.4.2 Reshaping only the traffic burst

Here, we consider the special case in which only the maximal burst size of the traffic can be reshaped along the path.

Suppose that, at each link $l \in p$, the traffic of a session $m$ is shaped with a fixed rate $\rho_m^l$, as at the entrance to the network. Furthermore, suppose that, at each link $l$, the traffic burst of session $m$ is reshaped with a parameter $\sigma_m^l$. Accordingly, at each link $l$, the traffic of session $m$ entering the EDF scheduler is $(\rho_m^l, \sigma_m^l)$-burstiness constrained.
Define the link-delay-burst function \( W_l(\sigma) \), for all \( 0 \leq \sigma \leq \sigma^m_0 \), as

\[
W_l(\sigma^m_i) = \{ \min d | F_l(t) - 1 (t - d) (\sigma^m_i + \rho^m_0 (t - d)) > 0, \forall t > 0 \}.
\]

\( W_l(\sigma^m_i) \) is the minimum delay that can be guaranteed to a session \( m \) at link \( l \), as a function of \( \sigma^m_i \). A typical instance of \( W_l(\sigma^m_i) \) (which corresponds to \( F_l(t) \) of Figure 3.1) is depicted in Figure 3.6. We note that \( W_l(\sigma^m_i) \) is an increasing piecewise linear function of \( \sigma^m_i \). Furthermore, the number of points at which the slope of the function \( W_l(\sigma^m_i) \) changes is less than \( 2I_l - l \), where \( I_l \) is the number of sessions at link \( l \) with a deadline of at most \( \delta^m_0 \). Denote the burst size instances for which the slope of \( W_l(\sigma^m_i) \) changes by \( x^l_i \) and let \( x^l_0 = 0 \). Additionally, denote the slope values and the function values of \( W_l(\sigma^m) \) at these points (starting with the right hand side) by \( a^l_i \) and \( y^l_i \), respectively. Let \( J_l \) be the number of points at which the slope changes, where \( J_l \leq 2I_l - l \).

Algorithm Calculate-Link-Delay-Burst, specified in Figure 3.7, calculates the set \( \{ y^l_j, x^l_j, a^l_j \}_{0 \leq j \leq J_l} \) for a link \( l \in E \), given the link parameters \( \{w^l_i, \delta^l_i, r^l_i\}_{0 \leq i \leq I_l} \) (the link’s work availability function) and given the pending session’s parameters \( (\rho^m_0, \sigma^m_0) \).

**Proposition 3.4** (a) Let \( I_l \) be a schedulable set of sessions at link \( l \), and let \( \{w^l_i, \delta^l_i, r^l_i\}_{0 \leq i \leq I_l} \) be a set of parameters defining the link work availability function. Also, let \( (\rho^m_0, \sigma^m_0) \) characterize a new session \( m \) such that the stability condition \( \sum_{i \in I_l} \rho^m_i + \rho^m_0 < R_l \) is satisfied. Then, the link-delay-burst \( W_l(\sigma^m) \), is fully specified by the set \( \{ y^l_j, x^l_j, a^l_j \}_{0 \leq j \leq J_l} \), obtained by Algorithm Calculate-Link-Delay-Burst.

(b) The algorithm’s complexity is \( O(I_l^2) \).

**Proof.** First, the algorithm finds the minimum guaranteed delay for a zero burst as follows.

Let \( (\alpha_i)_{i \in I_l} \) such that

\[
F_l(\delta^l_i) = \rho^m_0 (\delta^l_i - \alpha_i) \quad i \in I_l
\]
input: \{w_i^j, \delta_i^j, r_i^j\}_{0 \leq i \leq L}, (\rho_0^m, \sigma_0^m), \quad output: \{y_i^j, x_i^j, a_i^j\}_{0 \leq j \leq b}

1. \ c_a \leftarrow 0; \ c_\beta \leftarrow 0
2. \text{for} \ i=1 \ \text{to} \ I_l \ \text{do}
   (a) \text{if} \ w_i^j > 0 \ \text{then}
      i. \ \alpha_i \leftarrow \delta_i^j - \frac{w_i^j}{\rho_0^m}
      ii. \ c_\alpha \leftarrow \max(c_\alpha, \alpha_i)
   (b) \text{else} \ c_\beta \leftarrow \max(c_\beta, \delta_i^j)
3. \ c \leftarrow \max(c_\alpha, c_\beta)
4. \ b \leftarrow 0
5. \text{while} \ (b \leq I_l + 1) \ \text{and} \ (c \geq \delta_i^j) \ \text{do} \ b \leftarrow b + 1
6. \ j \leftarrow 0
7. \text{if} \ w_{b-1} = 0
   (a) \ y_i^j \leftarrow \delta_i^{b-1}; \ x_i^j \leftarrow 0; \ a_i^j \leftarrow \frac{1}{r_i^j}
   (b) \ c \leftarrow w_i^{b-1}; \ e \leftarrow b - 1
   (c) \text{for} \ k = b \ \text{to} \ I_l + 1 \ \text{do}
      i. \ \text{if} \ w_k^b - \rho_0^m (\delta_k^b - \delta_k^{b-1}) < c
         \ \text{then} \ c \leftarrow w_k^b - \rho_0^m (\delta_k^b - \delta_k^{b-1}); \ e \leftarrow k
   (d) \ c \leftarrow \frac{w_j^b - w_j^{b-1} + \delta_j^{b-1} - \delta_j^b}{r_j^{b-1}}
   (e) \ j \leftarrow j + 1
   (f) \ y_i^j \leftarrow c; \ x_i^j \leftarrow x_i^{j-1} + \frac{y_i^j - y_i^{j-1}}{a_i^{j-1}}; \ a_i^j \leftarrow \frac{1}{\rho_0^m}
8. \text{else} \ y_i^j \leftarrow c_\alpha; \ x_i^j \leftarrow 0; \ a_i^j \leftarrow \frac{1}{\rho_0^m}
9. \ b \leftarrow b + 1
10. \text{while} \ b < I_l + 1 \ \text{and} \ x_i^j < \sigma_0^m
    (a) \ c \leftarrow w_i^{b-1}; \ e \leftarrow b - 1
    (b) \text{for} \ k = b \ \text{to} \ I_l + 1 \ \text{do}
       i. \ \text{if} \ w_k^b - \rho_0^m (\delta_k^b - \delta_k^{b-1}) < c
          \ \text{then} \ c \leftarrow w_k^b - \rho_0^m (\delta_k^b - \delta_k^{b-1}); \ e \leftarrow k
       (c) \ \text{if} \ w_k^b - \rho_0^m (\delta_k^b - \delta_k^{b-1}) > w_i^{b-1} \ \text{do}
          i. \ j \leftarrow j + 1
          ii. \ y_i^j \leftarrow \delta_i^{b-1}; \ x_i^j \leftarrow x_i^{j-1} + \frac{y_i^j - y_i^{j-1}}{a_i^{j-1}}; \ a_i^j \leftarrow \frac{1}{r_i^j}
          iii. \ c \leftarrow \frac{w_i^b - w_i^{b-1} + \delta_i^{b-1} - \delta_i^b}{r_i^{b-1}}
          iv. \ j \leftarrow j + 1
          v. \ y_i^j \leftarrow c; \ x_i^j \leftarrow x_i^{j-1} + \frac{y_i^j - y_i^{j-1}}{a_i^{j-1}}; \ a_i^j \leftarrow \frac{1}{\rho_0^m}
    (d) \ b \leftarrow b + 1
11. \text{if} \ b = I_l + 1 \ \text{and} \ b \neq 1
    (a) \ j \leftarrow j + 1
    (b) \ y_i^j \leftarrow \delta_i^{b-1}; \ x_i^j \leftarrow x_i^{j-1} + \frac{y_i^j - y_i^{j-1}}{a_i^{j-1}}; \ a_i^j \leftarrow \frac{1}{r_i^{j-1}}
12. \ J_l \leftarrow j

Figure 3.7: Algorithm Calculate-Link-Delay-Burst

75
and let $c_\alpha = \max (\alpha_i, 0)$.

Let $c_\beta = \max_{i \in I_l : F_i(\delta) = 0} (\delta_i)$ and $c = \max (c_\alpha, c_\beta)$.

Let $b, 1 \leq b \leq I_l + 1$, such that $\delta_i^{b-1} \leq m < \delta_i^b$, where $\delta_i^0 = 0$ and $\delta_i^{I_l+1} = \infty$.

If $F_i(\delta_i^{b-1}) = 0$ then, the minimum guaranteed delay is $\delta_i^{b-1}$, i.e., $y_i^0 = \delta_i^{b-1}$. The slope of $W_i(\sigma^m)$ is $\frac{1}{\rho_i^0}$. Furthermore, let

$$e = \arg \min_{k > b-1} F_i(\delta_i^k) - \rho_i^m (\delta_i^k - \delta_i^{b-1})$$

and let $\alpha$ be such that

$$F_i(\delta_i^e) - \rho_i^m (\delta_i^e - \alpha) = F_i(\delta_i^{b-1}) + r_i^{b-1} (\alpha - \delta_i^{b-1}).$$

Then, the slope of $W_i(\sigma^m)$ changes at $W_i(\sigma^m) = \alpha$ to $\frac{1}{\rho_i^0}$.

Otherwise (i.e., $F_i(\delta_i^{b-1}) \neq 0$), the minimum guaranteed delay for a zero burst is $c_\alpha$ and the first slope of $W_i(\sigma^m)$ is $\frac{1}{\rho_i^0}$.

Next, the algorithm iterates over the delays $\{\delta_i^b\}_{b \leq I_l+1}$ in order to find the corresponding points at which the slope of $W_i(\sigma^m)$ changes.

Let

$$e = \arg \min_{k > b-1} F_i(\delta_i^k) - \rho_i^m (\delta_i^k - \delta_i^{b-1}).$$

If

$$F_i(\delta_i^e) - \rho_i^m (\delta_i^e - \delta_i^{b-1}) > F_i(\delta_i^{b-1})$$

then the slope of $W_i(\sigma^m)$ changes at $W_i(\sigma^m) = \delta_i^{b-1}$ to $\frac{1}{\rho_i^0}$. Furthermore, the slope of $W_i(\sigma^m)$ changes again at $W_i(\sigma^m) = c$ where $c$ is such that

$$F_i(\delta_i^e) - \rho_i^m (\delta_i^e - c) = F_i(\delta_i^{b-1}) + r_i^{b-1} (c - \delta_i^{b-1}).$$

Here, the slope changes to $\frac{1}{\rho_i^0}$.

For each $i, 1 \leq i \leq I_l$, the algorithm consists $O(I_l)$ operations. Thus, the algorithm’s complexity is $O(I_l^2)$.

Now, we have that the end-to-end delay along a path $\mathbf{p}$ is given by

$$D^m(\mathbf{p}) = \max_{l \in \mathbf{p}} \left( \frac{(\sigma_0^m - \sigma_l^m)^+}{\rho_0^m} \right) + \sum_{l \in \mathbf{p}} (W_l(\sigma_l^m) + d_l).$$

**Corollary 3.1** Suppose that the session $m$ guaranteed end-to-end delay $D^m(\mathbf{p})$ is minimized by the set $\{\bar{\sigma}_l^m\}_{l \in \mathbf{p}}$. Then, $\{\bar{\sigma}_l^m\}_{l \in \mathbf{p}}$ have the same value, i.e., $\bar{\sigma}_l^m = \bar{\sigma}^m \forall l \in \mathbf{p}$.
Proof. By contradiction. Assume that $\sigma^m_l = \hat{\sigma}^m$ $\forall l \in p$ does not hold. Then, let $\tilde{\sigma}^m = \min_{l \in p} \sigma^m_l$. Clearly, $\max_{l \in p} \left( \frac{(\sigma^m_l - \sigma^m)}{\rho^m_0} \right) = \max_{l \in p} \left( \frac{\sigma^m_0 - \sigma^m}{\rho^m_0} \right)$. However, since $W_l(\sigma^m_l)$ is increasing in $\sigma^m_l$, we have that $W_l(\tilde{\sigma}^m) \leq W_l(\hat{\sigma}^m) \forall l \in p$. Accordingly, the end-to-end delay along the path $p$ with a maximal burst size of $\tilde{\sigma}^m \forall l \in p$ is lower than the end-to-end delay with maximal burst sizes of $\hat{\sigma}^m \forall l \in p$, which contradicts the optimality of $\bar{\sigma}^m$. \hfill \blacksquare

Thus, aiming at minimizing the end-to-end delay, we seek a single maximal burst size value for all the links along the path.

Denote $W(\sigma^m) = \sum_{l \in p} W_l(\sigma^m)$. It is easy to see that $W(\sigma^m)$ is increasing and piecewise linear. Furthermore, if the slope of $W(\sigma^m)$ changes at $\sigma_0$ then for at least one link $l \in p$, say $l_0$, the slope of $W_{l_0}(\sigma^m)$ changes at $\sigma_0$. Accordingly, the slope of $W(\sigma^m)$ changes at most $2hI_{\max}$ times, where $I_{\max} = \max_{l \in p} I_l^-$ and $h$ is the number of hops along $p$.

Given a path $p$, the following corollary considers the optimal reshaping burst $\tilde{\sigma}^m$ that minimizes the guaranteed end-to-end delay $D^m(p)$.

**Corollary 3.2** Suppose that the session $m$ guaranteed end-to-end delay $D^m(p)$ is minimized by a maximal burst size $\bar{\sigma}^m$. Then, $W(\sigma^m)$ changes its slope at $\bar{\sigma}^m$.

**Proof.** By contradiction. Assume that $W(\sigma^m)$ is differentiable at $\bar{\sigma}^m$. Then, consider the following two possible cases.

Case 1: $W'(\bar{\sigma}^m) < \frac{1}{\rho^m_0}$.

Then, consider a maximal burst size $\tilde{\sigma}^m = \bar{\sigma}^m + \epsilon$. We have that

$$\frac{\sigma^m_0 - \sigma^m}{\rho^m_0} + W(\bar{\sigma}^m) + \sum_{l \in p} d_l < \frac{\sigma^m_0 - \sigma^m}{\rho^m_0} + W(\tilde{\sigma}^m) + \sum_{l \in p} d_l,$$

which contradicts the optimality of $\bar{\sigma}^m$.

Case 2: $W'(\bar{\sigma}^m) > \frac{1}{\rho^m_0}$.

Then, consider a maximal burst size $\tilde{\sigma}^m = \bar{\sigma}^m - \epsilon$. We have that

$$\frac{\sigma^m_0 - \sigma^m}{\rho^m_0} + W(\bar{\sigma}^m) + \sum_{l \in p} d_l < \frac{\sigma^m_0 - \sigma^m}{\rho^m_0} + W(\tilde{\sigma}^m) + \sum_{l \in p} d_l,$$

which contradicts the optimality of $\bar{\sigma}^m$. \hfill \blacksquare

Corollary 3.2 suggests that an optimal routing scheme may limit itself to consider reshaping burst sizes at which $W_l(\sigma^m)$, $\forall l \in E$, change their slopes. Accordingly, for each such burst size, Algorithm QFPTS-Fixed Rate (QFPTS-FR), depicted in Figure 3.8, finds the shortest path w.r.t. $W_l(\sigma^m) + d_l$. Then, among the $O(MI_{\max})$
1. for all $l \in E$, call Calculate-Link-Delay-Burst

2. for all values of $\sigma^m$, $\sigma^m \in \left\{ \{x_j^l\}_{0 \leq j \leq h_l} \right\}_{l \in E}$
   
   (a) for all $l \in E$, calculate $W_l(\sigma^m)$
   
   (b) find the shortest path $\tilde{p}$ between $s^m$ and $t^m$ w.r.t. $\{W_l(\sigma^m) + d_l\}$
       (through Dijkstra’s shortest path algorithm)

3. among the $O(MI_{\text{max}})$ paths, choose a path $\tilde{p}$ and burst $\tilde{\sigma}^m$ with the smallest guaranteed end-to-end delay
   
   $$D^m(p) = \frac{(\sigma^m_0 - \sigma^m_0)^+}{\rho^m_0} + \sum_{l \in \tilde{p}} (W_l(\sigma^m) + d_l).$$

4. if $D^m(\tilde{p}) < D^m$ then
   
   (a) $\delta_l^m \leftarrow W_l(\tilde{\sigma}^m)$, $\forall l \in \tilde{p}$
   
   (b) $(\tilde{\sigma}^m_l, \tilde{\rho}^m_l) \leftarrow (\sigma^m_l, \rho^m_0)$, $\forall l \in \tilde{p}$
   
   (c) return $\tilde{p}$, $\{\delta^m_l\}_{l \in \tilde{p}}$, $\{(\tilde{\sigma}^m_l, \tilde{\rho}^m_l)\}_{l \in \tilde{p}}$

5. else
   
   (a) there is no feasible path
   
   (b) end

Figure 3.8: Algorithm QFPTS-FR

paths, it chooses a path $\tilde{p}$ and burst $\tilde{\sigma}^m$ that minimize the guaranteed end-to-end delay.

**Proposition 3.5** Algorithm QFPTS-FR correctly solves QFPTS problem for a fixed reshaping traffic rate $\rho^m_0$. The algorithm’s complexity is $O(MI_{\text{max}}^2 + MI_{\text{max}}(N \log N + M))$.

**Proof.** The first part of the proposition follows from Corollaries 3.1 and 3.2. The algorithm consists of $M$ executions of Algorithm Calculate-Link-Delay-Burst as well as $O(MI_{\text{max}})$ executions of Dijkstra’s shortest path algorithm. The second part of the proposition immediately follows. ■

While the computational complexity of Algorithm QFPTS-FR is polynomial, it could still be prohibitively large. Therefore, in order to obtain an efficient yet computationally tractable solution, we establish an $\epsilon$-optimal solution with a complexity of $O(MI_{\text{max}}^2 + M (N \log N + M) \frac{1}{\epsilon} \log D_{\text{max}})$. To that end, we quantize the delay deadlines at each link.

Suppose that the allowed values of guaranteed link delay are restricted to a set of $O(\log_{1+\epsilon} D_{\text{max}})$ delay-classes, where $D_{\text{max}}$ is the maximal end-to-end delay
requirement. More precisely, at each link $l \in E$, the delay deadline assignment has to assume a value out of the set $0, 1 + \epsilon, (1 + \epsilon)^2, \ldots, (1 + \epsilon)^K$, where $K = \lceil \log_{1+\epsilon} D_{\text{max}} \rceil$. Accordingly, the work availability function at each link $l$ might change its slope only at the points $0, 1 + \epsilon, (1 + \epsilon)^2, \ldots, (1 + \epsilon)^K$. Consequently, the slope of the link-delay-burst function changes at most $2 \cdot K$ times. The corresponding algorithm QFPTS-FR-Delay Quantized - $\epsilon$ (QFPTS-FR-DQ-$\epsilon$), depicted in Figure 3.9, is similar to algorithm QFPTS-FR, only that it consists of $O \left( M \log_{1+\epsilon} D_{\text{max}} \right)$ iterations and it assigns deadlines out of the allowed set.

1. for all $l \in E$, call Calculate-Link-Delay-Burst
2. for all values of $\sigma^m$, $\sigma^m \in \{ \{ x^j_l \}_{0 \leq j \leq J_l} \}_{l \in E}$
   (a) for all $l \in E$, calculate $W_l(\sigma^m)$
   (b) find the shortest path $\tilde{p}$ between $s^m$ and $t^m$ w.r.t. $\{ W_l(\sigma^m) + d_l \}$ (through Dijkstra’s shortest path algorithm)
3. among the $O(MI_{\text{max}})$ paths, choose a path $\tilde{p}$ and burst $\tilde{\sigma}^m$ with the smallest guaranteed end-to-end delay $D^m(\tilde{p}) = \frac{(\sigma^m_{i_0} - \sigma^m_{i_1})}{\rho^m_{i_0}} + \sum_{l \in \tilde{p}} (W_l(\sigma^m) + d_l)$.
4. if $D^m(\tilde{p}) < D^m$ then
   (a) for each $l \in \tilde{p}$
     i. let $k_l$ such that $W_l(\tilde{\sigma}^m) > (1 + \epsilon)^{k_l-1}$ and $W_l(\tilde{\sigma}^m) < (1 + \epsilon)^{k_l}$
     ii. $\delta^m_l \leftarrow (1 + \epsilon)^{k_l}$
     iii. $(\tilde{\sigma}^m_l, \tilde{\rho}^m_l) \leftarrow (\tilde{\sigma}^m, \tilde{\rho}^m_l)$
   (b) return $\tilde{p}$, $\{ \delta^m_l \}_{l \in \tilde{p}}$, $\{(\tilde{\sigma}^m_l, \tilde{\rho}^m_l)\}_{l \in \tilde{p}}$
5. else
   (a) there is no feasible path
   (b) end

Figure 3.9: Algorithm QFPTS-FR-RQ-$\epsilon$

**Proposition 3.6** (a) The guaranteed end-to-end delay of the output path $\tilde{p}^m$ of algorithm QFPTS-FR-DQ-$\epsilon$ is at most $1 + \epsilon$ larger then the minimal value obtained by Algorithm QFPTS-FR, i.e., if $p^*$ is the output path of algorithm QFPTS-FR then $D^m(\tilde{p}^m) \leq (1 + \epsilon) \cdot D^m(p^*)$.
(b) The algorithm’s complexity is $O \left( MI_{\text{max}}^2 + M (N \log N + M) \frac{1}{\epsilon} \log D_{\text{max}} \right)$.
Proof. For each link \( l \in \tilde{p} \) \( \delta_l^m \) is at most \((1 + \epsilon) W_l (\tilde{\sigma}_m)\), i.e., \( \delta_l^m \leq (1 + \epsilon) W_l (\tilde{\sigma}_m) \). Accordingly,

\[
D^m (\tilde{p}) = \left( \frac{\sigma_0^m - \sigma^m}{\rho_0^m} \right) + \sum_{l \in \tilde{p}} (\delta_l^m + d_l) \\
\leq (1 + \epsilon) \left( \frac{1}{\rho_0^m} \left( \frac{\sigma_0^m - \sigma^m}{\rho_0^m} \right) + \sum_{l \in \tilde{p}} (W_l (\tilde{\sigma}_m) + d_l) \right) = (1 + \epsilon) D^m (p^*) .
\]

(3.3)

The algorithm consists of \( M \) executions of Algorithm Calculate-Link-Delay-Burst as well as \( O(M \log_{1+\epsilon} D_{\text{max}}) \) executions of Dijkstra’s shortest path algorithm. The second part of the proposition follows from the following relation:

\[
O \left( \log_{1+\epsilon} D_{\text{max}} \right) = O \left( \frac{\log D_{\text{max}}}{\log (1 + \epsilon)} \right) = O \left( \frac{1}{\epsilon} \cdot \log D_{\text{max}} \right) .
\]

\[\square\]

### 3.5 Optimizing the Path Selection

In order to supervise multiple connections throughout the network, the routing algorithm must consider the efficient use of the consumed resource. In the following, we devise two criteria for balancing the load, as well as the corresponding routing schemes.

#### 3.5.1 Rate Consumption Criterion

Consider first the simple criterion of choosing, for a connection request \( m \), a path \( p \) for which the residual rate (after establishing the new connection) of its bottleneck link is maximal. That is, we aim at solving the following problem.

**Maximum Residual Bottleneck (MRB):** Given are a network \( G (V,E) \), with a service rate \( R_l \), a propagation delay \( d_l \), and a work availability function \( F_l (t) \) for each \( l \in E \). Also, given is a session \( m \) with source \( s^m \), destination \( t^m \), upper rate \( \rho_0^m \), burst \( \sigma_0^m \) and an end-to-end delay requirement \( D^m \). Find a feasible path-deadline assignment \( (p^m, \delta^m) \), i.e., a path between \( s^m \) and \( t^m \) and a deadline allocation \( (\delta_l^m)_{l \in E} \), such that:

1. \( \sum_{l \in E} (\delta_l^m + d_l) \leq D^m \),

2. for all \( l \in E \):

\[
F_l (t) - 1 (t - \delta_l^m) (\sigma_0^m + \rho_0^m (t - \delta_l^m)) > 0 \quad \forall t > 0 .
\]
If there are several such paths, select the one with the maximum residual rate of its bottleneck, i.e., \( \max_{p} \min_{l \in p} (R'_l - \rho^m) \).

Let \( R^1, R^2, \ldots R^K \) be the set of all residual rate values, where \( K \leq M \). Assume, without the loss of generality, that \( R^1 > R^2 > \ldots > R^K \). Algorithm MRB, specified in Figure 3.10, solves the above problem.

1. for all \( l \in E \), calculate the minimum delay \( \delta^m_{l,0} \) that can be guaranteed to session \( m \) (through Algorithm MINIMUM-DELAY of [43])
2. find the shortest path \( \tilde{p} \) between \( s^m \) and \( t^m \) w.r.t. \( \{ \delta^m_{l,0} + d_l \} \) (through Dijkstra’s shortest path algorithm)
3. if \( \sum_{l \in \tilde{p}} (\delta^m_{l,0} + d_l) < D^m \) then
   (a) \( L \leftarrow 1, \ H \leftarrow K, \ k \leftarrow 1 \)
   (b) while \( k \neq \lceil \frac{L+H}{2} \rceil \)
      i. \( k \leftarrow \lfloor \frac{L+H}{2} \rfloor \)
      ii. delete all links \( l \) with \( R'_l < R^k \)
      iii. find the shortest path \( \tilde{p} \) w.r.t. \( \{ \delta^m_{l,0} + d_l \} \)
      iv. if \( \sum_{l \in \tilde{p}} (\delta^m_{l,0} + d_l) < D^m \) then \( H \leftarrow k \) else \( L \leftarrow k \)
   (c) \( \eta \leftarrow D - \sum_{l \in \tilde{p}} (\delta^m_{l,0} + d_l) \)
   (d) \( \delta^m_l \leftarrow \delta^m_{l,0} + \frac{\eta}{h} \), where \( h \) is the number of hops along the path \( \tilde{p} \)
   (e) return \( \tilde{p}, \{ \delta^m_l \}_{l \in \tilde{p}} \)
4. else there is no feasible path, end

Figure 3.10: Algorithm MRB

**Proposition 3.7** Algorithm MRB correctly identifies a feasible path (if one such exists) with the maximum residual rate of its bottleneck link. The algorithm’s complexity is \( O(M I_{\text{max}} + \log K (N \log N + M)) \).

**Proof.** The first part of the proposition is straightforward.

The algorithm calculates MINIMUM-DELAY for all links, which involves \( O(M I_{\text{max}}) \) operations. Then, the algorithm consists \( O(\log K) \) executions of Dijkstra’s shortest path algorithm, which involves \( O(\log K (N \log N + M)) \) operations. \( \blacksquare \)
3.5.2 Delay-Rate Consumption Criterion

The way that a session affects the availability of resources at a link depends on the residual rate as well as the assigned deadline at that link. In other words, a session $m$ that is routed through a link $l$ consumes more resources when it is assigned a smaller deadline. While algorithm MRB balances the rate consumption, the path selection does not account for the allocation of the delay deadlines. More precisely, the MRB scheme balances the deadline assignment along the path $\hat{p}$ only after the path is selected. Accordingly, we now consider an alternative routing scheme, which aims at balancing the load as well as optimizing the deadline allocation. To that end, a new criterion for load balancing is called for.

More specifically, we seek a measure for the residual resource available for future connections after admitting a connection. Such a measure should account for the rate consumption as well as the deadline allocation.

Recall that the EDF schedulability condition at each link $l \in E$ is given by the inequality $F_l(t) > 0$, $\forall t > 0$. Thus, when admitting a new connection $m$ with a deadline $\delta_l^m$, the following inequality should hold:

$$\min_{t > \delta_l^m} (F_l(t) - 1(t - \delta_l^m)(\sigma_0^m + \rho_0^m(t - \delta_l^m))) > 0.$$  

Accordingly, for all $d > \delta_l^m$, define the residual resource function as follows:

$$S_l(d) = \min_{t > d} (F_l(t) - 1(t - d)(\sigma_0^m + \rho_0^m(t - d)))$$

$S_l(d)$ estimates the residual resources available for future connections as a function of the deadline assignment for the newly established connection. In other words, consider a connection established at two links $l_1$ and $l_2$ with deadlines $\delta_{l_1}$ and $\delta_{l_2}$, and suppose that $S_{l_1}(\delta_{l_1}) < S_{l_2}(\delta_{l_2})$; then, the connection establishment has a more severe impact on link $l_1$ than on link $l_2$ in terms of reducing the ability to admit future connection requests. Intuitively, $S_l(d)$ can be found by sliding the function $F_l(t)$ from its leftmost position $\delta_l^m$ to the right (see Figure 3.11). For each point (i.e., for each $d$), the value of $S_l(d)$ is given by the minimum distance between $F_l(t)$ and $1(t - d)(\sigma_0^m + \rho_0^m(t - d))$. A typical
instance of $S_l(d)$ is depicted in Figure 3.11. Note that $S_l(d)$ is increasing and piecewise-linear with alternating slopes. Denote the delay instances for which the slope of $S_l(d)$ changes by $x^l_i$ and let $x^l_i = \delta^m_i \delta^0_i$. Additionally, denote the slope values and the function values of $S_l(d)$ at these points (starting with the right hand side) by $\rho^l_i$ and $y^l_i$, respectively. Let $J$ be the number of points at which the slope changes. Then, the residual-resource function is given by the set $(y^l_i, x^l_i, \rho^l_i)_{0 \leq j \leq J}$.

Algorithm Calculate-Residual-Resource, specified in Figure 3.12, calculates the set $(y^l_i, x^l_i, \rho^l_i)_{0 \leq j \leq J}$ for a link $l \in E$, given the link parameters $(w^l_i, \delta^l_i, r^l_i)_{0 \leq i \leq i}$, (the link’s work availability function) and given the pending session’s parameters $(\rho^m_i, \sigma^m_i)$. First, the algorithm considers a zero traffic burst. Then, at step 6, the algorithm makes the necessary modifications to account for a traffic burst of $\sigma^m_0$. Assuming a zero burst, the algorithm considers all the minima of the work availability function $F_l(t)$. For each minimum, it identifies the points at which the slope changes. More specifically, suppose there is a minimum at time $\delta^l_i$, which corresponds to the deadline of session $i$; then step 2b identifies whether, for all $d$ in the range $\delta^l_i < d \leq \delta^l_i$, the minimum $\min_{t > d} F_l(t) - 1(t - d)(\rho^m_0 \cdot t)$ is obtained at time $t = \delta^l_i$. If so, then step 2c is executed. At that step, the algorithm identifies a potential point at which the slope changes to $\rho^m_0$, and a second point at which the slope changes to $r^l_i$. Otherwise, at step 2b only one potential point is identified, namely the one at which the slope changes to $\rho^m_0$.

**Proposition 3.8** (a) Let $\mathcal{I}_l$ be a schedulable set of sessions at link $l$, and let $\{w^l_i, \delta^l_i, r^l_i\}_{0 \leq i \leq i}$ be a set of parameters defining the link work availability function. Also, let $(\rho^m_0, \sigma^m_0)$ characterize a new session $m$ such that the stability condition $\sum_{i \in \mathcal{I}_l} \rho^l_i + \rho^m_i < R_l$ is satisfied. Then, the link residual resource function $S_l(d)$, is fully specified by the set $(y^l_i, x^l_i, \rho^l_i)_{0 \leq j \leq J}$, obtained by Algorithm Calculate-Residual-Resource.

(b) The algorithm’s complexity is $O(|\mathcal{I}_l|)$.

**Proof.** Algorithm Calculate-Residual-Resource iterates over the delays $\{\delta^l_i\}_{i \in \mathcal{I}_l}$ and identifies the corresponding points at which the slope of $S_l(d)$ changes. The algorithm implements the following scheme.

For each $i \in \mathcal{I}_l$, let

$$e = \arg \min_{k > i} F_l\left(\delta^l_k - \rho^m_0 (\delta^l_k - \delta^l_i)\right).$$

If $F_l(\delta^l_i) - \rho^m_0 (\delta^l_i - \delta^l_e) \leq F_l(\delta^l_i)$ then find $x$ such that

$$F_l\left(\delta^l_i - \rho^m_0 (\delta^l_i - \delta^l_e) - r^l_{i} - 1 \left(x - \delta^l_i - 1\right) = F_l(\delta^l_i) - \rho^m_0 (\delta^l_i - x).ight.$$
input: \{w_i^k, \delta_i^k, r_i^k\}_{0 \leq i \leq I}, (\rho_0^m, \sigma_0^m)

output: \{y_j^l, x_i^j, a_i^j\}_{0 \leq j \leq J_i}

1. \(k \leftarrow 0, x_0^0 \leftarrow 0, a_0^0 \leftarrow r_0^0, y_0^0 \leftarrow 0\)
2. for \(i = 1\) to \(I:\)
   (a) \(z_i^i \leftarrow \begin{cases} \min w_i^j - \rho_0^m (\delta_i^k - \delta_i^l), & i < I_l \\ w_i^i, & i = I_l \end{cases}\)
   (b) if \(z_i^i \leq w_i^i\)
      i. \(x \leftarrow \frac{z_i^i - w_i^i - \rho_0^m \delta_i^k + r_i^i - \delta_i^l}{\delta_i^k - \delta_i^l}\)
      ii. if \(\delta_i^k - 1 < x < \delta_i^l\) then
          A. \(k \leftarrow k + 1\)
          B. \(\bar{x}_i^k \leftarrow x, \bar{a}_i^k \leftarrow \rho_0^m, \bar{y}_i^k \leftarrow y_i^k + \bar{a}_i^k - 1 (x_i^k - \bar{x}_i^{k-1})\)
   (c) else if \(z_i^i > w_i^i\)
      i. \(x \leftarrow \frac{w_i^i - w_i^i - \rho_0^m \delta_i^k + r_i^i - \delta_i^l}{\delta_i^k - \delta_i^l}\)
      ii. if \(\delta_i^l - 1 < x < \delta_i^k\) then
          A. \(k \leftarrow k + 1\)
          B. \(\bar{x}_i^k \leftarrow x, \bar{a}_i^k \leftarrow \rho_0^m, \bar{y}_i^k \leftarrow y_i^k + \bar{a}_i^k - 1 (x_i^k - \bar{x}_i^{k-1}),\)
          iii. \(k \leftarrow k + 1\)
          iv. \(\bar{x}_i^k \leftarrow \delta_i^k, \bar{a}_i^k \leftarrow r_i^k, \bar{y}_i^k \leftarrow y_i^k + \bar{a}_i^k - 1 (x_i^k - \bar{x}_i^{k-1})\)
3. \(k \leftarrow k + 1\)
4. \(\bar{x}_i^k \leftarrow \delta_i^k, \bar{a}_i^k \leftarrow r_i^k, \bar{y}_i^k \leftarrow w_i^l\)
5. \(K \leftarrow k\)
6. if \(\bar{y}_i^K \leq \sigma_0^m\)
   (a) \(x_0^0 \leftarrow \bar{x}_i^K + \frac{\sigma_0^m - \bar{y}_i^K}{\bar{a}_i^K}, a_0^0 \leftarrow \bar{a}_i^K, y_0^0 \leftarrow 0\)
   (b) \(J_i \leftarrow 0\)
7. else
   (a) let \(b\) such that \(\bar{y}_i^{b-1} \leq \sigma_0^m\) and \(\bar{y}_i^b > \sigma_0^m\).
   (b) \(j \leftarrow 0, x_i^0 \leftarrow \bar{x}_i^k - \frac{\bar{y}_i^b - \sigma_0^m}{\bar{a}_i^b - 1}, a_i^0 \leftarrow \bar{a}_i^{b-1}, y_i^0 \leftarrow 0\).
   (c) for \(k \leftarrow b\) to \(K:\)
      i. \(j \leftarrow j + 1, x_i^j \leftarrow \bar{x}_i^k, a_i^j \leftarrow \bar{a}_i^k, y_i^j \leftarrow \bar{y}_i^k - \sigma_0^m\)
   (d) \(J_i \leftarrow j\)

Figure 3.12: Algorithm Calculate-Residual-Resource
if $\delta_i^{-1} < x < \delta_i$ then the slope of $S_l(d)$ changes at $x$ (see, for example, point $x_1^2$ in Figure 3.13(a)). Otherwise $(F_l(\delta_i) - \rho_{0m}(\delta_i - \delta_i)) > F_l(\delta_i)$, find $x$ such that

$$F_l(\delta_i^{-1}) + r_i^{-1}(x - \delta_i^{-1}) = F_l(\delta_i) - \rho_{0m}(\delta_i - x);$$

if $\delta_i^{-1} < x < \delta_i$ then the slope of $S_l(d)$ changes at $x$ (see, for example, point $x_1^2$ in Figure 3.13(b)). Furthermore, the slope of $S_l(d)$ changes again at $\delta_i$ (see, for example, point $x_3^2$ in Figure 3.13(b))

![Figure 3.13: Typical instances of the residual resource function](image)

One can see that the above scheme identifies all the points at which the slope of $S_l(d)$ changes as well as the function and the slope values at these points. Accordingly, the corresponding set $\{y_i^l, x_i^l, a_i^l\}_{0 \leq j \leq I_l}$, identified by the algorithm, fully specifies $S_l(d)$.

For each $i$, $1 \leq i \leq I_l$, the algorithm consists $O(I_l)$ operations. Thus, the algorithm’s complexity is $O(I_l^2)$.

One can see that the way by which the path selection and deadline allocation affect the available resources at the network links depends on the the value of the residual resource function $S_l(\delta_l)$. Therefore, a better measure for balancing the loads over the network might be one that accounts for the residual resources, rather than the residual rate. Accordingly, we aim at balancing the load by seeking a feasible path for which the residual resource of its bottleneck link is maximal. That is, we seek a path that solve the following problem.

**Maximum Residual Resource (MRR):** Given are a network $G(V, E)$, with a service rate $R_l$, a propagation delay $d_l$, and a work availability function $F_l(t)$ for each $l \in E$. Also, given is a session $m$ with source $s^m$, destination $t^m$, upper rate $\rho_{0m}$, burst $\sigma_{0m}$ and an end-to-end delay requirement $D^m$. Find a feasible path-deadline assignment $(p^m, \delta^m)$, i.e., a path between $s^m$ and $t^m$ and a deadline allocation $(\delta_l)_{l \in p^m}$, such that:
1. \[ \sum_{l \in \mathbf{P}^m} (\delta_l^m + d_l) \leq D^m, \]

2. for all \( l \in \mathbf{P}^m \):
   \[ F_l(t) - 1 (t - \delta_l^m) (\sigma_0^m + \rho_0^m (t - \delta_l^m)) > 0 \quad \forall t > 0. \]

If there are several such paths, select the path-deadline assignment that solves the following:
\[ \max_{\mathbf{P}, (\delta_l)_{l \in \mathbf{P}}} \min_{l \in \mathbf{P}} (S_l(\delta_l)). \]

Algorithm MRR, specified in Figure 3.14, identifies such a path. First the algorithm calculates for each link \( l \in E \) the residual-resource function through algorithm Calculate-Residual-Resource (see sketch in Figure 3.15). Then, it performs a binary search to find the maximal minimal allowed residual-resource value for which a feasible path exists. For each minimal allowed value, the algorithm calculates the minimum available deadline \( \delta_l^m \) at each link \( l \) (through the residual-resource function). Then, it finds the shortest path with respect to \( \{\delta_l^m + d_l\} \). If the identified path is feasible, then the minimal residual-rate value can be increased, otherwise it should be decreased.

**Proposition 3.9** Algorithm MRR correctly solves the problem. The algorithm’s complexity is
\[ O \left( M \cdot I_{\text{max}}^2 + \log (R'_{\text{max}} D) \left( N \log N + M \right) \right). \]

**Proof.** The first part of the proposition results from the fact that the residual-resource function is increasing.

The algorithm calculates the residual-resource for all links, which involves \( O (MI_{\text{max}}^2) \) operations. Then, the algorithm consists \( O (\log (R'_{\text{max}} D)) \) iterations; at each iteration Dijkstra’s shortest path algorithm is executed.

\[ \square \]

### 3.6 Simulation Results

We now demonstrate the efficiency of the proposed load-balancing routing schemes by way of simulations. Our figure of merit is session blocking probability, which is evaluated for various loads and network topologies.

Following [43], the characteristics of the sessions that need to be supported in the network are generated randomly and are intended to cover a wide range of traffic patterns. We take \( \rho = 10^p \text{Kb/s} \), where \( p \) is uniformly distributed in \([1,3]\). Hence, \( \rho \) covers the range \([10 \text{Kb/s}, 1 \text{Mb/s}]\). Next, we take \( \sigma = r \cdot \rho \text{Kb} \), where \( r \) is uniformly distributed in \([0.8, 1.6]\). Session requests are generated according to a Poisson process with a parameter \( \alpha \). The session’s source and destination nodes are uniformly chosen. A session is accepted if the considered routing scheme identifies
a feasible route; otherwise, the session is rejected. An accepted session stays in the system for an exponentially distributed duration time with mean $\frac{1}{\beta}$. The ratio $\frac{\alpha}{\beta}$ characterizes the traffic load offered to the network, i.e., the average number of flows that would exist at any time at a network with no capacity limitation. The blocking probability is evaluated as the total number of rejected sessions divided by the total number of generated sessions.

Each session has a delay requirement $D = 10^s \cdot 30ms$, where $s$ is uniformly.

1. for all $l \in E$, call Calculate-Residual-Resource
2. for all $l \in E$, let the minimum delay $\delta_{l,0}^m \leftarrow x_{l}^0$
3. find the shortest path $\bar{p}$ between $s^m$ and $t^m$ w.r.t. $\{\delta_{l,0}^m + d_l\}$ (through Dijkstra’s shortest path algorithm)
4. if $\sum_{l \in \bar{p}} (\delta_{l,0}^m + d_l) < D$ then
   a. for all $l \in E$
      i. if $D > x_{l}^{K_l}$ then $R_l(D) \leftarrow y_{l}^{K_l} + a_{l}^{K_l}(D - x_{l}^{K_l})$
      ii. else let $b_l$ such that $x_{l}^{b_l-1} \leq D < x_{l}^{b_l}$
         A. $R_l(D) \leftarrow y_{l}^{b_l-1} + a_{l}^{b_l-1}(D - x_{l}^{b_l-1})$
   b. $L \leftarrow 0$ $H \leftarrow \max_{l} R_l(D)$, $\lambda \leftarrow H$
   c. while $|\lambda - \frac{L+H}{2}| > 1$
      i. $\lambda \leftarrow \frac{L+H}{2}$
      ii. for all $l \in E$
         A. if $\lambda > y_{l}^{K_l}$ then $\delta_l^m \leftarrow x_{l}^{K_l} + \frac{\lambda - y_{l}^{K_l}}{a_{l}^{K_l}}$
         B. else let $b_l$ such that $y_{l}^{b_l-1} \leq \lambda < y_{l}^{b_l}$,
            $\delta_l^m \leftarrow x_{l}^{b_l-1} + \frac{\lambda - y_{l}^{b_l-1}}{a_{l}^{b_l-1}}$
         iii. find the shortest path $\bar{p}$ between $s^m$ and $t^m$ w.r.t. $\{\delta_{l}^m + d_l\}$ (through Dijkstra’s shortest path algorithm)
   iv. if $\sum_{l \in \bar{p}} (\delta_{l}^m + d_l) \leq D$ then $L \leftarrow \lambda$ else $H \leftarrow \lambda$
   d. return $\bar{p}$, $\{\delta_{l}^m\}_{l \in \bar{p}}$
5. else there is no feasible path, end

Figure 3.14: Algorithm MRR
1. for all $l \in E$, compute the residual resource function
2. find the shortest path $\tilde{p}$ between $s^m$ and $t^m$ w.r.t. $\{d^m_{l,0} + \delta_l\}$ (through Dijkstra’s shortest path algorithm)
3. if $\sum_{l \in \tilde{p}} (d^m_{l,0} + \delta_l) < D$ then
   (a) for all $l \in E$ compute $S_l(D)$
   (b) $L \leftarrow 0$ $H \leftarrow \max_l S_l(D)$, $\lambda \leftarrow H$
   (c) while $|\lambda - \frac{L+H}{2}| > 1$
      i. $\lambda \leftarrow \frac{L+H}{2}$
      ii. for all $l \in E$, compute the minimum $d^m_l$ such that $S_l(d^m_l) \geq \lambda$
      iii. find the shortest path $\tilde{p}$ between $s^m$ and $t^m$ w.r.t. $\{d^m_l + \delta_l\}$ (through Dijkstra’s shortest path algorithm)
     iv. if $\sum_{l \in \tilde{p}} (d^m_l + \delta_l) \leq D$ then $L \leftarrow \lambda$ else $H \leftarrow \lambda$
   (d) return $\tilde{p}$, $\{d^m_l\}_{l \in \tilde{p}}$
4. else there is no feasible path, end

Figure 3.15: Algorithm MRR (sketch)

distributed in $[0, 1.52]$, thus $D$ ranging in $[30ms, 1s]$. We conducted the simulations on the network topologies depicted in Figure 3.16, which were extensively considered in network performance studies.

![Network Topologies](image.png)

Figure 3.16: Network topologies

We compared between the following routing algorithms:

- Our maximum residual bottleneck algorithm (MRB), which identifies a feasible
path with the maximum residual rate of its bottleneck link.

- Our maximum residual resource algorithm (MRR), which identifies a feasible path with the maximum residual resource of its bottleneck link.

- The quickest path algorithm, which identifies a feasible path with the minimum end-to-end delay bound.

- The minimum hop path algorithm, which identifies a feasible path with the minimum number of hops.

Figure 3.17: Session request blocking probability

We generated a total of 10,000 sessions in each simulation run, i.e., for each routing scheme and each traffic load. Each simulation run was repeated 5 times with different seeds. Figure 3.17 presents the simulation results, in terms of the blocking probability as a function of the traffic load, in the two network topologies. One can see that our maximum residual resource (MRR) and maximum residual bottleneck (MRB) routing schemes outperform both the quickest path as well as the minimum hop path routing schemes. In particular, focusing on a load region for which the blocking percentage is 10% (i.e., a significant yet still reasonably acceptable value), our maximum residual resource (MRR) algorithm admits about 60% more sessions than the quickest path routing scheme in the COST239 topology, and about 80% more sessions in the Clustered topology. Furthermore, Algorithm MRR admits over 100% more sessions than the minimum-hop path routing scheme, in both topologies.

3.7 Conclusion

This study has considered QoS routing in networks that employ the rate-controlled EDF scheduling discipline.
First, we focused on the basic problem of identifying feasible paths. Here, we broadened the space of feasible solutions by allowing to reshape the traffic with different parameters at each hop. Accordingly, we established an optimal routing scheme that considered the joint problem of identifying a feasible path and optimizing the reshaping parameters along the path. However, the computational complexity of this scheme might be prohibitively large. Thus, we established an approximation scheme that is based on quantizing the reshaping delay, which is $\epsilon$-optimal and at the same time computationally efficient. Our scheme is guaranteed to find a feasible solution (i.e., path) whenever the standard EDF routing schemes do so. Moreover, we demonstrated that our scheme identifies feasible solutions also in cases in which the standard schemes fail. We also considered the special case in which only the maximal burst size of the traffic can be reshaped along the path. For this case, we established an optimal routing scheme as well as an $\epsilon$-optimal solution of lower complexity.

Next, we considered the more complex problem, of optimizing the route choice in terms of balancing the loads and accommodating multiple connections. Here, we established and validated two routing schemes. The first scheme aims at balancing the load by identifying a feasible path (if one exists) with a maximum bottleneck residual rate. The second scheme jointly considers the problems of route selection and assignment of deadlines along the chosen path. Clearly, considering the problems jointly, instead of independently as in previous approaches, leads to a better utilization of the network resources. Simulation results demonstrated the advantages of our schemes, in particular the second one.

To sum up, in this chapter we have investigated two novel classes of EDF routing schemes: the first allows per-hop optimization of the reshaping parameters, while the second aims to optimize the global utilization of network resources. An important yet complex task, left for future research, is to merge the two classes, i.e., establish an efficient EDF routing scheme that performs per-hop reshaping optimization and optimizes the utilization of the network.
Chapter 4

The EDF scheduling discipline - stochastic setting

4.1 Introduction

In this chapter, we consider the Rate-Controlled Non-Preemptive Earliest-Deadline-First (RC-NPEDF) scheduling discipline [7]. Some important properties have been established for this scheduling policy, in the context of deterministic settings. In [27], exact schedulability conditions have been established; these conditions detect violations of delay guarantees in a network of EDF switches. In [40], NPEDF was proven to be an optimal scheduling discipline; that is, if a set of tasks is schedulable under any non-preemptive scheduling discipline, then this set is also schedulable under NPEDF. Also, RC-NPEDF was proven to outperform GPS in providing end-to-end delay guarantees in a network [4].

As mentioned, the above results have been established solely within a deterministic setting. In view of the potential practical advantages of the RC-NPEDF service discipline, in this study we investigate QoS provision through RC-NPEDF scheduling for connections with stochastic traffic profiles and stochastic QoS requirements. Specifically, we consider Exponentially Bounded Burstiness (EBB) processes, which appropriately model typical input processes [13]. First, we study the single node case. We derive schedulability conditions for the NPEDF scheduling discipline and establish its optimality in terms of schedulable regions. Furthermore, we derive an upper bound on the tail distribution of the delay experienced by a packet entering an NPEDF scheduling element. Moreover, we introduce the concept of EBB traffic shapers, and derive an upper bound on the tail distribution of the delay experienced by any packet entering such a traffic shaper. Next, we study the multiple node (i.e., multi-hop path) case, and derive an upper bound on the end-to-end delay tail distribution.

We then show that the guaranteed upper bound on the end-to-end delay tail distribution in RPPS-GPS networks can be guaranteed in RC-NPEDF networks as
Some previous studies also considered QoS provisioning under a stochastic setting [33, 44, 45]. However, those studies dealt with deterministic input traffic and deterministic traffic shaping (dual-leaky-bucket). Our stochastic model and framework are different. In our case, the stochasticity of the end-to-end guarantees is (solely) due to the stochastic nature of the session input traffic, whereas in [33, 44, 45], the exploitation of statistical multiplexing results in a provision of (only) stochastic guarantees even for deterministically bounded input traffic.

The rest of the chapter is structured as follows. In Section 4.2, we formulate the model. Next, in Section 4.3 we consider a service element in isolation. Then, in section 4.4, we study the multiple node case and derive a stochastic bound on the end-to-end delay. In Section 4.5, we conclude the chapter and discuss possible future work. To maintain presentation continuity, some of the proofs are omitted from the main text and gathered as an appendix in Section 4.6.

### 4.2 Model Formulation

We consider a store-and-forward network comprising of packet switches, in which a packet scheduler is available at each output link. Packetized traffic from a particular connection entering the switch passes through a traffic shaper before being delivered to the scheduler. The traffic shaper regulates traffic, so that the output of the shaper satisfies certain pre-specified traffic characteristics. We consider the EBB (Exponentially Bounded Burstiness) traffic model. The traffic shaper reshapes the incoming traffic by delaying the packets so that the output is EBB, and then delivers them to the scheduler. We focus on the Non-Preemptive Earliest Deadline First (NPEDF) scheduling discipline. For ease of presentation we shall henceforth use the notation EDF instead of NPEDF (Non-Preemptive EDF). An EDF scheduler associates a deadline \( t + \delta^j \) with each packet of a session \( j \) that arrives at time \( t \). The packets are served non-preemptively in the order of their assigned deadlines.

The network is represented by a directed Graph \( G(V, E) \), in which nodes represent switches and arcs represent links. \( V \) is the set of nodes and \( E \) is the set of links interconnecting them. Each link \( l \in E \) is characterized by a service rate \( r_l \). We assume that link propagation delays are negligible. Let \( L^j \) be the maximal packet size of a session \( j \) and \( L_{\text{max}} \) be the maximal packet size in the network.

Let \( A(t) \) be the instantaneous traffic rate. This rate is a stochastic process, expressing the instantaneous intensity of the data flow in a link. The amount of data flowing in this traffic stream from any time instant \( s \) to any (later) time instant \( \tau \) is denoted by \( A[s, \tau] \). We assume a discrete time domain, in which the amount of information transmitted on a link with capacity \( r = 1 \) during one time slot is
regarded as a unit of data. In this context, we have $A[s, \tau] = \sum_{n=s+1}^{\tau} A(n)$.

### 4.3 The single node case

We start by considering the basic single node case, namely, a (single) traffic shaper followed by a (single) EDF scheduler. First, we consider a Non-Preemptive EDF scheduling element in isolation. Here, we generalize the notion of EDF-schedulability in order to apply to the considered stochastic setting, and derive the corresponding schedulability conditions. With that at hand, we establish the optimality (in terms of schedulable regions) of the EDF discipline among the class of non-preemptive scheduling disciplines. Then, we consider EBB traffic and establish an upper bound on the tail distribution of the delay experienced by a packet entering a single EDF scheduler. Next, we introduce the notion of EBB traffic shapers. Here, we establish a rule for delaying packets in the traffic shaper so that the traffic delivered to the scheduler conforms with the required EBB characteristics.

#### 4.3.1 EDF Scheduler

An Earliest-Deadline-First (EDF) scheduler assigns each arriving packet a time stamp corresponding to its deadline, i.e., a packet from connection $j$ with a deadline $\delta^j$ that arrives at the scheduler at time $t$ is assigned a time stamp of $t + \delta^j$. The EDF scheduler always selects the packet with the lowest deadline for transmission. Under the deterministic setting, if a packet is not transmitted by its time stamp then a deadline violation has occurred. A set of connections is said to be schedulable if deadline violations never occur. Obviously, the above definitions cannot apply to the stochastic setting.

**Schedulability conditions**

Consider a set $\mathcal{N}$ of sessions where each session traffic rate is $\{A_j(t)\}_{j \in \mathcal{N}}$. Each session $j$ requires some stochastic delay guarantees. Assume that each session is associated with a certain probability $q^j$, which reflects its “sensitivity” to delay violations. Then, a session $j$ traffic entering a scheduler is guaranteed an effective delay $\gamma^j$ if the delay violation probability is at most $q^j$. More formally,

**Definition 4.1** Given are a scheduler and a set $\mathcal{N}$ of sessions entering the scheduler. A session $j \in \mathcal{N}$ is guaranteed an effective delay $\gamma^j$ with respect to $q^j$ if for all $t$ $\Pr \{D^j(t) \geq \gamma^j\} \leq q^j$, where $D^j(t)$ is the actual scheduling delay suffered by session $j$ at time $t$. 

93
Accordingly, the schedulability of a set of sessions entering an EDF scheduler is given in the following definition.

**Definition 4.2** Given are a scheduler and a set $\mathcal{N}$ of sessions. Each session $j \in \mathcal{N}$ is characterized by a stochastic traffic profile $A^j \{t, \tau\}$ and a required effective delay $\gamma^j$ with delay violation of at most $q^j$. The set of sessions is said to be schedulable if, for all $t > 0$, the required effective delay can be guaranteed for all sessions.

**Proposition 4.1** Consider a set $\mathcal{N}$ of sessions entering a non-preemptive EDF scheduler. Then, we have for all $t$:

$$P r \{D^j(t) \geq \delta^j + d\} \leq \sum_{\hat{\tau} = 0}^{t} Pr \left\{ \sum_{j \in \mathcal{N}} A^j \{t - \hat{\tau}, t - \delta^j\} + \max_{\delta^j > \hat{\tau}} L^j \geq r (\hat{\tau} + d) \right\}$$

(4.1)

**Proof.** See Section 4.6.1. ■

**Corollary 4.1** Let $\gamma^1 \leq \gamma^1 \leq \ldots \leq \gamma^N$. Any vector $\{\gamma^1, \gamma^2, \ldots, \gamma^{|\mathcal{N}|}\}$ is schedulable under EDF if for all $j \in \mathcal{N}$ and for all $t$:

$$\frac{L_{\text{max}}}{r} \leq \gamma^1$$

(4.2)

and

$$\sum_{\hat{\tau} = 0}^{t} Pr \left\{ \sum_{j \in \mathcal{N}} A^j \{t - \hat{\tau}, t - \gamma^j\} + \max_{\gamma^j > \hat{\tau}} L^j \geq r \hat{\tau} \right\} \leq q^j$$

(4.3)

**Proof.** The corollary follows immediately from the definition of schedulability and from Proposition 4.1 by assigning to each session $j$ a deadline $\delta^j = \gamma^j$ and taking $d = 0$. ■

The following lemma establishes necessary conditions for a set of sessions $\mathcal{N}$ to be schedulable under any non-preemptive policy.

**Lemma 4.1** Let $\gamma^1 \leq \gamma^1 \leq \ldots \leq \gamma^N$. If the effective delay vector $\{\gamma^1, \gamma^2, \ldots, \gamma^{|\mathcal{N}|}\}$ is schedulable under a non-preemptive policy then, for all $j \in \mathcal{N}$ and for all $t$, it holds that

$$\frac{L_{\text{max}}}{r} \leq \gamma^1$$

(4.4)

and

$$\sum_{\hat{\tau} = 0}^{t} Pr \left\{ \sum_{j \in \mathcal{N}} A^j \{t - \hat{\tau}, t - \gamma^j\} + \max_{\delta^j > \hat{\tau}} L^j \geq r \hat{\tau} \right\} \leq q^j$$

(4.5)
Proof. See Section 4.6.2.

With the above results, the following theorem states that EDF is optimal, in terms of schedulable regions, among the class of non-preemptive policies.

**Theorem 4.1** Let $\gamma^1 \leq \gamma^1 \leq \ldots \leq \gamma^N$. If all sessions $j, 1 \leq j \leq N$, has stochastic traffic rate $A_j^j(t)$, then the EDF scheduling discipline is optimal among the class of non-preemptive scheduling discipline and their schedulable region consists of the set of effective delay vectors $\{\gamma^1, \gamma^2, \ldots, \gamma^N\}$ that satisfy the following constraints for all $j \in N$ and for all $t$:

$$\frac{L_{\text{max}}}{r} \leq \gamma^j \quad (4.6)$$

$$\sum_{\tau=0}^{t} \text{Pr} \left\{ \sum_{j \in N} A_j^j [t - \tau, t - \gamma^j] + \max_{\gamma^j > \tau} L_j^j \geq r^\tau \right\} \leq q^j \quad (4.7)$$

*Proof.* Follows from Corollary 4.1 and Lemma 4.1.

**An example: two exponentially distributed sessions**

Through a simple example, we compare the schedulable regions of the EDF and GPS scheduling disciplines when the input traffic has exponentially bounded bursts. More precisely, we consider two input sessions and assume that their traffic has exponentially distributed bursts. For both the EDF and GPS scheduling disciplines, we derive the schedulable regions based on the schedulability conditions in Theorem 4.1 above and in [6], respectively.

Consider two sessions ($i = 1, 2$) entering a scheduler. Assume that packet size is negligible and let the traffic burst be exponentially distributed, that is, for $i = 1, 2$ and for all $t$:

$$\text{Pr} \left\{ A^i(t) \geq \rho^i t + \sigma \right\} = e^{-\alpha^i \sigma}.$$

We assume that the service rate is $r = 1$. We consider two sets of input traffic: (i) symmetric traffic with parameters $\rho^1 = 0.44, \alpha^1 = 1.01, \rho^2 = 0.44, \alpha^2 = 1$, $q^1 = q^2 = 10^{-5}$ (Figure 4.1(a)) and (ii) asymmetric traffic with parameters $\rho^1 = 0.6, \alpha^1 = 2, \rho^2 = 0.2, \alpha^2 = 0.5, q^1 = q^2 = 10^{-5}$ (Figure 4.1(b)).

The schedulable region for the EDF scheduling discipline is calculated according to Theorem 4.1. For the GPS scheduling discipline, we have by the upper bound on the delay tail distribution in [5, 6] that the sessions are schedulable if there exist $r^1, r^2$ ($r^1 + r^2 \leq r$) such that, for $i = 1, 2$, it holds that

$$\frac{1}{1 - e^{-\alpha^i (r^i - \rho^i)}} e^{-\alpha^i r^i \delta^i} \leq q^i.$$
From both figures one can see that, indeed, the schedulable region of the EDF discipline contains the schedulable region of the GPS discipline. Moreover, there are effective delay values (e.g., $13 \leq \gamma^1 \leq 24$, $13 \leq \gamma^1 \leq 24$, for the symmetric case, and $7 \leq \gamma^1 \leq 10, 26 \leq \gamma^1 \leq 69$, for the asymmetric case) that can be guaranteed under the EDF discipline and cannot be guaranteed under the GPS discipline.

**Upper bound on the delay tail distribution**

With the schedulability conditions at hand, we consider EBB traffic and show that the delay experienced by a packet entering an EDF scheduler is exponentially bounded (i.e., EB).

**Proposition 4.2** Suppose that $\{A^j\}_{j \in N}$ are $|N|$ independent $(\rho^j, \Lambda^j, \alpha^j)$-EBB processes sharing an EDF server with delay assignment $\{\delta^j\}_{j \in N}$. Then, at any time $t$, for any $d > 0$ and for all $k \in N$,

$$\Pr \{D^k(t) \geq \delta^k + d\} \leq \Gamma e^{-\gamma d},$$

where $\frac{1}{\alpha} = \sum_{j \in N} \frac{1}{\alpha^j}$, $\Gamma = \frac{\sum_{j \in N} \Lambda^j e^{-\alpha^j \rho^j (\delta^j - \max)}}{1 - \frac{1}{\Gamma} e^{-\alpha (\delta^k - \max)}}$ and

$$\gamma = \hat{\alpha} \left( r - \sum_{j \in N} \rho^j \right) + \min_{j \in N} \alpha^j \rho^j.$$
Proof. From the proof of Proposition 4.1 (expression (4.32)), we have

\[
\Pr \{ D^k(t) \geq \delta_k + d \} \leq \sum_{\hat{t}=0}^t \Pr \left\{ \sum_{j \in \mathcal{N}} A_j^i \left[ t - \hat{t}, t + \delta_k - \delta_j \right] + \max_{\delta_j > \hat{t} + \delta_k} L_j^i > r(\hat{t} + \delta_k + d) \right\}. \quad (4.9)
\]

Let \( p^1, p^2, \ldots, p^{|\mathcal{N}|} \) be positive constants that sum to 1. Then,

\[
\left\{ \sum_{j \in \mathcal{N}} A_j^i \left[ t - \hat{t}, t + \delta_k - \delta_j \right] + L_{\text{max}} \geq r(\hat{t} + \delta_k + d) \right\} \subset \bigcup_{j \in \mathcal{N}} \left\{ A_j^i \left[ t - \hat{t}, t + \delta_k - \delta_j \right] \geq p^j \left( r(\hat{t} + \delta_k + d) - L_{\text{max}} \right) \right\} \quad (4.10)
\]

Recall that the session’s traffic is EBB, i.e.,

\[
\Pr \{ A_j^i \left[ t - \hat{t}, t + \delta_k - \delta_j \right] \geq p^j \left( \hat{t} + \delta_k - \delta_j \right) + \sigma^j \} \leq \Lambda_j e^{-\alpha^j \sigma^j}, \quad \forall \hat{t} > \delta_j - \delta_k. \quad (4.11)
\]

Let \( \mathcal{N}^1 \subset \mathcal{N} \) be the set of sessions for which \( \delta_j > \delta_k \), and let \( \mathcal{N}^2 = \mathcal{N} - \mathcal{N}^1 \). Then, from (4.9), (4.10), (4.11) and the union bound, we get

\[
\Pr \{ D^k(t) \geq \delta_k + d \} \leq \sum_{j \in \mathcal{N}^1} \sum_{\hat{t} = \delta_j - \delta_k}^t \Lambda_j e^{-\alpha^j \left( p^j \left( r(\hat{t} + \delta_k + d) - L_{\text{max}} \right) - \rho^j \left( \hat{t} + \delta_k - \delta_j \right) \right)}
\]

\[
+ \sum_{\hat{t} = 0}^t \sum_{j \in \mathcal{N}^2} \Lambda_j e^{-\alpha^j \left( p^j \left( r(\hat{t} + \delta_k + d) - L_{\text{max}} \right) - \rho^j \left( \hat{t} + \delta_k - \delta_j \right) \right)}
\]

\[
\leq \sum_{j \in \mathcal{N}^1} \Lambda_j e^{-\alpha^j \left( p^j \left( r(\hat{t} + \delta_k + d) - L_{\text{max}} \right) - \rho^j \left( \delta_k - \delta_j \right) \right)} \sum_{\hat{t} = \delta_j - \delta_k}^\infty e^{-\alpha^j \left( p^j r \hat{t} - \rho^j \hat{t} \right)}
\]

\[
+ \sum_{j \in \mathcal{N}^2} \Lambda_j e^{-\alpha^j \left( p^j \left( r(\hat{t} + \delta_k + d) - L_{\text{max}} \right) - \rho^j \left( \delta_k - \delta_j \right) \right)} \sum_{\hat{t} = 0}^\infty e^{-\alpha^j \left( p^j r \hat{t} - \rho^j \hat{t} \right)}
\]

\[
\leq \sum_{j \in \mathcal{N}^1} \frac{\Lambda_j}{1 - e^{-\alpha^j (p^j r - \rho^j)}} e^{-\alpha^j \left( p^j r (\hat{t} + d) - p^j L_{\text{max}} \right)}
\]

\[
+ \sum_{j \in \mathcal{N}^2} \frac{\Lambda_j}{1 - e^{-\alpha^j (p^j r - \rho^j)}} e^{-\alpha^j \left( p^j r (\hat{t} + d) - p^j L_{\text{max}} - \rho^j (\delta_k - \delta_j) \right)}
\]

\[
\leq \sum_{j \in \mathcal{N}} \frac{\Lambda_j}{1 - e^{-\alpha^j (p^j r - \rho^j)}} e^{-\alpha^j \left( \left( p^j r - \rho^j \right) (\hat{t} + d) - p^j L_{\text{max}} + \rho^j (\delta_j + d) \right)} ,
\]

97
where the set \( \{ p^j \} \) is such that \( p^j r > \rho^j \), \( \sum_{j \in N} p^j = 1 \). Let \( p^j \) be such that \( \alpha^j (p^j r - \rho^j) = c \) \( \forall j \in N \). Accordingly, we have that \( c = \tilde{\alpha} \left( r - \sum_{j \in N} \rho^j \right) \). Then,

\[
\Pr \left\{ D^k(t) \geq \delta^k + d \right\} \\
\leq \sum_{j \in N} \frac{\Lambda^j}{1 - e^{-\tilde{\alpha} \left( r - \sum_{j \in N} \rho^j \right)}} e^{\left( r - \sum_{j \in N} \rho^j \right) + \alpha^j \rho^j} \frac{L_{max}}{r} e^{-\alpha^j \rho^j (\delta^k + d)} e^{-\tilde{\alpha} \left( r - \sum_{j \in N} \rho^j \right) (\delta^k + d)} \\
= \sum_{j \in N} \frac{\Lambda^j e^{-\alpha^j \rho^j (\delta^k + d - L_{max})}}{1 - e^{-\tilde{\alpha} \left( r - \sum_{j \in N} \rho^j \right)}} e^{-\tilde{\alpha} \left( r - \sum_{j \in N} \rho^j \right) (\delta^k + d - L_{max})} \\
\leq \sum_{j \in N} \frac{\Lambda^j e^{-\alpha^j \rho^j (\delta^k - L_{max})}}{1 - e^{-\tilde{\alpha} \left( r - \sum_{j \in N} \rho^j \right)}} e^{-\tilde{\alpha} \left( r - \sum_{j \in N} \rho^j \right) (\delta^k + d - L_{max})} e^{-\min_{j \in N} \alpha^j \rho^j d}
\]

and the proposition follows.

4.3.2 EBB traffic shaper

The EBB traffic shaper has one input link and one output link. The shaper receives a stream on the input link and it buffers data, if necessary, so that the output stream has EBB with the required parameters.

Suppose that an EBB traffic \( A_0 \) with parameters \( (\rho_0, \Lambda_0, \alpha_0) \) enters to a shaper \( A_1 \). Let \( A_1 \) represent the rate of the traffic exiting the shaper. Let \( s_i \) be the time at which the \( i \)th packet starts to arrive on the input link, and \( L_i \) is the length in bits of the packet. Suppose that the packet exits on the output link at time \( t_i \). Shaper \( A_1 \) transmits packets on the output link in FCFS order such that

\[
\Pr \left\{ W_\rho (A_1) (t_i) \geq \sigma \right\} \leq \Lambda_1 e^{-\alpha_1 \sigma}.
\]

Denote by \( D_i(A_0, A_1) \) the delay suffered by the \( i \)-th packet, \( D_i(A_0, A_1) = t_i - s_i \).

Proposition 4.3 Let

\[
D_i (A_0, A_1) = \frac{1}{\rho} \left( W_\rho (A_0) (s_i) - \left[ \beta_1 W_\rho (A_0) (s_i) - \varepsilon_1 \right]^+ \right). \tag{4.12}
\]
where $\rho > \rho_0$, $0 < \beta_1 \leq 1$, $\varepsilon_1 \geq 0$, and $W_\rho (A_0) (s_i)$ is defined as follows:\footnote{Note that $W_\rho (A_0) (s_i)$ is the size of the backlog at time $s_i$ in a virtual work-conserving system, which accepts traffic at the rate $A_0$ and transmits at the rate $\rho$. This value is known at the time $s_i$ that the $i$th packet arrives.}:

$$W_\rho (A_0) (s_i) = \max \{ A_0 (t, s_i) - \rho (s_i - t) \}. \quad (4.13)$$

Then, it holds that:

1. $W_\rho (A_1) (t_i) \leq (\beta_1 W_\rho (A_0) (s_i) - \varepsilon_1)^+$

2. for all $\sigma > 0$, $s > t > 0$, $\Pr \{ A_1 (t, s) \geq \rho (s - t) + \sigma \} \leq \Lambda_1 e^{-\alpha_1 \sigma}$, where $\Lambda_1 = \frac{\Lambda_0}{1 - e^{-\alpha_0(s-r_0)}} e^{-\alpha_1 \varepsilon_1}$ and $\alpha_1 = \frac{\alpha_0}{\beta_1}$.

3. $\Pr \{ D_i (A_0, A_1) \geq d \} \leq \frac{\Lambda_0}{1 - e^{-\alpha_0(s-r_0)}} e^{\alpha_0 \frac{\varepsilon_1}{\beta_1}} e^{-\alpha_0 \frac{\sigma}{\varepsilon_1}}$.

Proof. To prove Proposition 4.3, we first show that for all $i \geq 1$,

$$W_\rho (A_1) (t_i) = W_\rho (A_0) (s_i) - \rho D_i (A_0, A_1). \quad (4.14)$$

We establish (4.14) by induction. It holds for the first packet ($i = 1$) since $W_\rho (A_1) (t_1) = W_\rho (A_0) (s_1) = D_i (A_0, A_1) = 0$. We now assume that (4.14) holds for some fixed $i$ and show that it also holds for $i + 1$, i.e., we show that

$$W_\rho (A_1) (t_{i+1}) = W_\rho (A_0) (s_{i+1}) - \rho D_{i+1} (A_0, A_1).$$

Notice that $D_i (A_0, A_1) \leq \frac{W_\rho (A_0) (s_i)}{\rho}$. Then, it holds that

$$W_\rho (A_1) (t_{i+1}) = W_\rho (A_1) (t_i) + L_i - \rho (t_{i+1} - t_i)$$

$$= W_\rho (A_0) (s_i) - \rho D_i (A_0, A_1) + L_i - \rho (t_{i+1} - t_i)$$

$$= W_\rho (A_0) (s_i) + L_i - \rho (s_{i+1} - s_i) - \rho D_{i+1} (A_0, A_1)$$

$$= W_\rho (A_0) (s_{i+1}) - \rho D_{i+1} (A_0, A_1). \quad (4.15)$$

and (4.12) follows. From (4.12) and (4.14), one can easily see that

$$W_\rho (A_1) (t_i) = (\beta_1 W_\rho (A_0) (s_i) - \varepsilon_1)^+.$$

Considering $\sigma > 0$, it is easy to see that

$$\Pr \{ (\beta_1 W_\rho (A_0) (s_i) - \varepsilon_1)^+ \geq \sigma \} = \Pr \{ \beta_1 W_\rho (A_0) (s_i) - \varepsilon_1 \geq \sigma \}. \quad (4.16)$$
Recall that $A_0$ is EBB with the parameters $(\rho_0, \Lambda_0, \alpha_0)$. Thus, from Theorem 1 in [13], it follows that

$$ \Pr \{ W_\rho (A_0) (s_i) \geq \sigma \} \leq \frac{\Lambda_0}{1 - e^{-\alpha_0 (\rho - \rho_0)}} e^{-\alpha_0 \sigma}. \quad (4.17) $$

From (4.16) and (4.17), it follows that

$$ \Pr \{ W_\rho (A_1) (t_i) \} \leq \frac{\Lambda_0}{1 - e^{-\alpha_0 (\rho - \rho_0)}} e^{-\alpha_0 \sigma}. $$

The second part of the proposition follows from Theorem 1 in [13].

We have that

$$ \Pr \{ D_i(A_0, A_1) \geq d \} = \Pr \left\{ \frac{1}{\rho} (W_\rho (A_0) (s_i) - [\beta_1 W_\rho (A_0) (s_i) - \varepsilon_1]^+) \geq d \right\} $$

Consider the following cases.

**case 1**: $\beta_1 W_\rho (A_0) (s_i) \leq \varepsilon_1$. Then, $D_i(A_0, A_1) = \frac{1}{\rho} W_\rho (A_0) (s_i)$. Thus, from (4.17), we have

$$ \Pr \{ D_i(A_0, A_1) \geq d \} \leq \frac{\Lambda_0}{1 - e^{-\alpha_0 (\rho - \rho_0)}} e^{-\alpha_0 \rho d}. \quad (4.18) $$

**case 2**: $\beta_1 W_\rho (A_0) (s_i) < \varepsilon_1$. Then, we have

$$ \Pr \{ D_i(A_0, A_1) \geq d \} \leq \frac{\Lambda_0}{1 - e^{-\alpha_0 (\rho - \rho_0)}} e^{\alpha_0 \varepsilon_1} \frac{\varepsilon_1}{\beta_1} e^{-\alpha_0 \frac{d}{\beta_1}}. \quad (4.19) $$

From (4.18) and (4.19), we have

$$ \Pr \{ D_i(A_0, A_1) \geq d \} \leq \begin{cases} 
\frac{\Lambda_0}{1 - e^{-\alpha_0 (\rho - \rho_0)}} e^{-\alpha_0 \rho d} \quad & d \leq \frac{\varepsilon_1}{\beta_1} \\
\frac{\Lambda_0}{1 - e^{-\alpha_0 (\rho - \rho_0)}} e^{\alpha_0 \varepsilon_1} \frac{\varepsilon_1}{\beta_1} e^{-\alpha_0 \frac{d}{\beta_1}} \quad & d > \frac{\varepsilon_1}{\beta_1} \rho 
\end{cases} $$

and the third part of the Proposition follows.

We note that the EBB traffic shaper requires the knowledge of the backlog size $W_\rho (A_0) (s_i)$ at all packet arrival times. In the single node case, the backlog size can be calculated from $A_0$. However, in the multiple node case, which is considered next, the backlog size is available only at the first hop (since $A_0$ is available only at the entrance to the network). Therefore, we assume that the backlog size $W_\rho (A_0) (s_i)$ is added to the header of each packet.
4.4 The multiple node case

We proceed to consider a sequence of service elements, which constitute a path in the network.

4.4.1 Traffic shaper elements in series

First, we study the effect of connecting a sequence of some \( n \) \((\rho, \Lambda, \alpha)\)-shaper elements with the same rate, say \( \{(\rho, \Lambda_1, \alpha_1), (\rho, \Lambda_2, \alpha_2), \ldots, (\rho, \Lambda_n, \alpha_n)\} \). Let \((\rho, \Lambda_k, \alpha_k)\) be the parameters of the \( k \)-th shaper \( A_k \). \( A_k \) represents the rate of the traffic output of the \( k \)-th shaper and \( A_0 \) represents the rate of the traffic input to the system. Let \( D_i(A_0, A_{1:k}) \) be the difference between the time at which the \( i \)-th packet begins to exit the \( k \)-th shaper \((t^k_i)\) and the time at which it begins to arrive to the system \((s_i)\). Finally, let \( s^k_i \) be the time at which the \( i \)-th packet begins to arrive to the \( k \)-th shaper.

**Proposition 4.4** Let

\[
D_i(A_k, A_{k+1})(s^k_i) = \frac{1}{\rho} \left[ W_\rho(A_k)(s^k_i) - [\beta_{k+1} W_\rho(A_0)(s_i) - \varepsilon_{k+1}]^+ \right]. \tag{4.20}
\]

For all \( k = 1, 2, \ldots, n \) it holds that:

1. \( W_\rho(A_k)(t^k_i) = W_\rho(A_0)(s_i) - \rho D_i(A_0, A_{1:k}) \)

2. \( W_\rho(A_n)(t^n_i) \leq [\beta_k W_\rho(A_0)(s_i) - \varepsilon_k]^+ \)

3. for all \( \sigma > 0, s > t > 0 \), \( \Pr \{ A_n(t, s) \geq \rho(s - t) + \sigma \} \leq \Lambda_k e^{-\alpha_k \sigma} \),
   where \( \Lambda_k = \frac{\Lambda_0}{1 - e^{-\alpha_0(\rho - \rho_0)}} e^{-\alpha_k \varepsilon_k} \) and \( \alpha_k = \frac{\alpha_0}{\beta_k} \).

4. \( \Pr \{ D_i(A_0, A_{1:k}) \geq d \} \leq \max_{m=1, 2, \ldots, k} \frac{\Lambda_0}{1 - e^{-\alpha_0(\rho - \rho_0)}} e^{\alpha_0 \frac{c_m}{1+m} - \alpha_0 \frac{\rho d}{1+m}} e^{-\alpha_0 \frac{c_m}{1+m}}. \)

**Proof.** See Section 4.6.3.

In particular, Proposition 4.4 implies that if all the traffic shapers have the same parameters, say \((\rho, \Lambda_1, \alpha_1)\), then, the tail distribution of the shaping delay of the cascaded shapers is exponentially bounded with the same parameters as the shaping delay of a single \((\rho, \Lambda_1, \alpha_1)\)-shaper.

4.4.2 EDF-Scheduler and EBB-Shaper in Cascade

Our goal is to derive an upper bound on the tail distribution of the delay experienced by a packet entering a system of a (single) EDF-scheduler followed by a (single) EBB-shaper. To that end, we start by considering two systems, \( S_1 \) and \( S_2 \), where
$S_1$ consists of a $(\rho, \Lambda, \alpha)$-shaper and $S_2$ consists of a “delay subsystem” and an identical $(\rho, \Lambda, \alpha)$-shaper connected in series, as depicted in Figure 4.2. The “delay subsystem” delays the $i$th arriving packet by $\theta_i \geq 0$ and then delivers it to the shaper. The following lemma relates the delays experienced by a packet in the two systems $S_1$ and $S_2$. More precisely, the lemma states that the delay distribution in $S_2$ is upper-bounded by the same function that upper bounds the distribution of the delay in $S_1$.

![Diagram](image)

**Figure 4.2: The Systems $S_1$ and $S_2$**

**Lemma 4.2** Assume that packets arrive to systems $S_1, S_2$ according to the same arrival process $A(t)$. If $D_i^{(1)}$ and $D_i^{(2)}$ are the delays of packet $i$ in the traffic shaper in $S_1$ and $S_2$ respectively, then, for all $i = 1, 2, \ldots$,

$$D_i^{(1)} \leq D_i^{(2)} + \theta_i \quad (4.21)$$

**Proof.** See Section 4.6.4. □

The following proposition provides the required bound, namely on the delay distribution of an EDF-scheduler followed by an EBB-shaper.

**Proposition 4.5** Assume that an EBB traffic $A_0(t)$ with parameters $(\rho_0, \Lambda_0, \alpha_0)$ enters a $(\rho, \Lambda_1, \alpha_1)$-shaper $A_1$. Then, the output of $A_1$ enters a system $S$, for which it is known that the delay experienced by a packet $i$ is exponentially bounded as follows:

$$\Pr \{ D_i(t) \geq \delta + d \} \leq \Gamma e^{-\gamma d} \quad \forall d > 0. \quad (4.22)$$

The output of system $S$ enters a $(\rho, \Lambda_1, \alpha_1)$-shaper $A_2$.

The total delay, $\hat{D}_i$, experienced by a packet $i$, from the time it enters the scheduler (system $S$) till the time it exits $A_2$, is exponentially bounded as follows:

$$\Pr \{ \hat{D}_i \geq \delta + d \} \leq \Gamma e^{-\gamma d} \quad \forall d > 0. \quad (4.22)$$

**Proof.** Let $D_i$ be the delay of packet $i$ in system $S$, and let $D_i^{(1)}$ be its delay in $A_2$. Therefore, $\hat{D}_i = D_i + D_i^{(1)}$. First, consider a modified system where a delay subsystem with $\theta_i = (\delta - D_i)^+$ is inserted between $S$ and $A_2$, and let $D_i^{(2)}$ be the
delay of packet \( i \) in \( A_2 \) under this new arrangement. Applying Lemma 4.2 we have that \( D_i^{(1)} \leq \theta_i + D_i^{(2)} \).

Next, replace \( S \) with a modified system \( S' \) in which all packets experience the same delay \( D' \) such that

\[
\Pr \{ D' \geq \delta + d \} \leq \Gamma e^{-\gamma d} \quad \forall d > 0.
\]  

Figure 4.3: Original and Modified Systems

(See Figure 4.3.) Now, observe that, since the delay of every packet between its entrance to \( S' \) and its exit from the delay system is \( \max \{ D', \delta \} \), the traffic entering the shaper \( A_2 \) is a time-shifted version of the traffic exiting \( A_0 \).

Hence, \( D_i^{(2)} = D_i (A_1, A_2) \) and since \( A_1 \) and \( A_2 \) have the same shaping parameters, we have from proposition 4.4 that \( D_i (A_1, A_2) = 0 \). Accordingly, we have that \( \hat{D}_i \leq \max \{ \delta, D' \} \). Thus,

\[
\Pr \{ \hat{D}_i \geq \delta + d \} \leq \Pr \{ \max \{ \delta, D' \} \geq \delta + d \} = \Pr \{ D' \geq \delta + d \} \leq \Gamma e^{-\gamma d} \quad \forall d > 0.
\]

4.4.3 End-to-end Delay

Finally, consider a connection \( k \) with \((\rho_0^k, \Lambda_0^k, \alpha_0^k)\)-EBB input traffic. The connection is routed through a path \( p \) in which all nodes employ the RC-EDF service discipline. The traffic is reshaped along the path with parameters \((\rho^k, \Lambda^k, \alpha^k)\). Then, the stochastic bound on the end-to-end delay is given in the following theorem.

**Theorem 4.2** For any session \( k \) in an RC-EDF network with \((\rho_0^k, \Lambda_0^k, \alpha_0^k)\)-EBB input traffic and \((\rho^k, \Lambda^k, \alpha^k)\) traffic shapers along a path \( p \), the end-to-end delay \( D^k (p) \) tail distribution is upper bounded as follows:

\[
\Pr \{ D^k (p) \geq d \} \leq \left( \Lambda^k + \sum_{l \in p} \Gamma_l^k \right) e^{-\frac{d}{\lambda^k + \sum_{l \in p} \gamma_l^k}}
\]  

(4.24)
where

\[
\tilde{\Lambda}^k = \frac{\Lambda_0^k}{1 - e^{-\alpha} \left( (\rho_k - \rho_k^0) \right)} \left( \frac{\Lambda_0^k}{1 - e^{-\alpha} \left( (\rho_k - \rho_k^0) \right)} \right), \quad \Lambda^k = \frac{\alpha_0^k \rho_k}{\alpha_1^k - \alpha_0^k} \quad \text{and} \quad \Gamma^k \text{ as well as } \gamma^k_l \text{ are given in proposition 4.2.}
\]

Proof. The end-to-end delay consists of the delays along a series of shapers (with the same traffic parameters) and schedulers in cascade. According to Proposition 4.5, the end-to-end delay can be calculated from the delay in a the first shaper and the delays in a sequence of schedulers. In proposition 4.3, we showed that the delay in the shaper is EB, whereas in proposition 4.2 we showed that the delay in a single EDF scheduler is EB as well. Thus, the theorem follows from the calculus of EB processes [13].

4.4.4 Achieving RPPS GPS Guarantees

In [6], per-session upper bounds on the end-to-end delay tail distribution are developed for an RPPS GPS networks with EBB input traffic. We shall show that a rate-controlled service discipline can be designed to guarantee the same delay bounds.

To simplify the discussion, we shall assume a continuous flow model, i.e., with no packetization effects.

Consider first the design of traffic shapers. Consider a traffic shaper that reshapes the input traffic such that the output stream is burstiness constrained with an upper rate \( g \) and zero burst.

Lemma 4.3 Let \( A_0(t) \) be an EBB input traffic stream with parameters \( (\rho_0, \Lambda_0, \alpha_0) \). \( A_0(t) \) enters an EBB shaper with \( \rho = g \), \( \beta_1 \to 0 \) and \( \epsilon_1 = 0 \). Then, the output stream is burstiness constrained with an upper rate \( g \) and zero burst. Furthermore, the shaping delay tail distribution is upper bounded as follows:

\[
\Pr \{ D(t) \geq d \} \leq \frac{\Lambda}{1 - e^{-\alpha(g - \rho)} e^{-\alpha gd}}. \quad (4.25)
\]

Proof. Follows from proposition 4.3.

Now, consider a set of sessions \( \mathcal{N} \) entering a network and routed through several links. Denote the set of session entering link \( l \) by \( \mathcal{N}_l \). Let a path \( p(j) \) denote the set of links traversed by a session \( j \).

First, consider a GPS network, that is, a network that employ the GPS scheduling discipline at the output links of all nodes. Denote the service rate of an output link \( l \) by \( r_l \). Let \( g_l^j \) be the service rate guaranteed by the GPS scheduler to a session
j at the output link l, and let \( g^j \) be the guaranteed service rate along the path of that session, i.e., \( g^j = \min_{l \in p(j)} g^l \). From [6], we have that the tail distribution of the end-to-end delay of session \( j \), \( D^j(p) \), is upper bounded as follows:

\[
\Pr\{D^j(p) \geq d\} \leq \frac{N^j}{1 - e^{-\alpha^j(g^j - \rho^j)}} e^{-\alpha^j g^j d}.
\]

(4.26)

Next, consider an RC-EDF network. To achieve the above upper bound on the end-to-end delay tail distribution, we assume that the traffic shapers in the network are such that, for all \( j \in \mathcal{N} \), the output traffic of a session \( j \) is burstiness-constrained with an upper rate \( g^j \) and zero burst. Since the sessions are schedulable under the RPPS-GPS scheduling discipline (i.e., \( \forall l : \sum_{l \in \mathcal{N}_l} g^l < r_l \)), they are schedulable under the RC-EDF discipline as well. Notice that, with the above traffic shapers, the traffic at the entrance of each scheduler in the network is deterministically bounded with zero bursts. Thus, from [4], we have that such a traffic does not suffer from any delay in the network. Consequently, the end-to-end delay of a session \( j \) is solely composed of the shaping delay at the first link along its path. Therefore, with RC-EDF networks, EBB traffic and the above traffic shapers, we have from Lemma 4.3 that the end-to-end delay tail distribution is upper bounded as follows:

\[
\Pr\{D^j(p) \geq d\} \leq \frac{N^j}{1 - e^{-\alpha^j(g^j - \rho^j)}} e^{-\alpha^j g^j d},
\]

i.e., the same upper bound on the delay tail distribution (4.26) that is guaranteed in RPPS-GPS networks.

4.5 Conclusion

The study presented in this chapter considered end-to-end QoS provisioning in RC-EDF networks with stochastic traffic profiles and stochastic guarantees. Previous studies have dealt with either the generalized processor sharing scheduling discipline (GPS), under both deterministic and stochastic settings (e.g., [6, 13]), or with the rate-controlled earliest deadline first discipline (RC-EDF), under a deterministic setting (e.g., [4, 27, 40]). The present study is the first to provide end-to-end bounds for exponentially bounded burstiness (EBB) traffic and systems of RC-EDF schedulers. A second important contribution is the establishment of the optimality of the RC-NPEDF scheduling discipline in terms of schedulable regions under the stochastic setting.

After having established the bounds analytically, we tested them through some simulations and numerical calculations, carried on a three-node example. In the
example, the obtained bound for the RC-EDF service discipline practically outperformed the bound presented in [6] for the RPPS-GPS service discipline under EBB traffic. We then showed that any upper bound on the end-to-end delay tail distribution that can be guaranteed by the GPS discipline, can also be guaranteed by an RC-EDF discipline, by reshaping the traffic.

This chapter provides the required foundations for several related call admission control and routing problems. These problems are an important direction for future research.

4.6 Appendix

4.6.1 Proof of Proposition 4.1

Proposition 4.1: Consider a set $\mathcal{N}$ of sessions entering a non-preemptive EDF scheduler. Then, we have for all $t$:

$$
Pr \left\{ D^j (t) \geq \delta^j + d \right\} \leq \sum_{\tau = 0}^{t} \Pr \left\{ \sum_{j \in \mathcal{N}} A^j [t - \tau, t - \delta^j] + \max_{\delta^j > \tau} L^j \geq r (\tau + d) \right\}
$$

(4.27)

Proof. Consider a tagged packet from connection $k \in \mathcal{N}$ that arrives at the scheduler at time $t$ and is completely transmitted at time $t + D^k (t)$. Let $W^{k,t} (t + \tau)$ be the workload in the scheduler at time $t + \tau$ that is served before the tagged packet (from connection $k$ with arrival time $t$). The tagged packet has a deadline violation if for all $\tau$, $0 \leq \tau \leq \delta^k$, $W^{k,t} (t + \tau) > 0$, i.e.,

$$
Pr \{ D^k (t) \geq \delta^k + d \} = Pr \left\{ \min_{0 \leq \tau \leq \delta^k + d} \{ W^{k,t} (t + \tau) \} \geq 0 \right\}.
$$

Let $t - b (t)$ (where $b (t) \geq 0$) be the last time before $t$ at which the scheduler does not contain traffic with a deadline less than or equal to the deadline of the tagged packet. Since the scheduler is empty before time $0$, the time $t - b (t)$ is guaranteed to exist. $b (t)$ is given by:

$$
b (t) = \min \left\{ z \mid W^{\leq t + \delta^k} (t - z) = 0, z \geq 0 \right\},
$$

where $W^{\leq z} (y)$ denotes the workload in the scheduler at time $y$ due to packets with
deadlines less than or equal to \( x \). We have:

\[
\Pr \{ D^k (t) \geq \delta^k + d \} = \sum_{\hat{\tau} = 0}^{\tau} \Pr \left( \{ D^k (t) \geq \delta^k + d \} \cap \{ b(t) = \hat{\tau} \} \right)
\]

\[
= \sum_{\hat{\tau} = 0}^{\tau} \Pr \left( \left\{ \min_{0 \leq \tau \leq \delta^k + d} \{ W^{k,t} (t + \tau) \} \geq 0 \right\} \cap \{ b(t) = \hat{\tau} \} \right)
\]

\[
(4.28)
\]

Note that \( \{ b(t) = \hat{\tau} \} \) implies that, in each of the \( \hat{\tau} \) time slots, \( r \) units of data have been transmitted. Also, note that \( \{ W^{k,t} (t + \tau) \geq 0 \} \forall \tau \ (0 \leq \tau \leq \delta^k + d) \) implies that some additional \( r \cdot \tau \) data units have been transmitted from \( t \) to \( t + \tau \). Moreover, \( \{ b(t) = \hat{\tau} \} \) also implies that the queue at time \( t - \hat{\tau} \) did not contain any packets whose deadline was less than or equal to \( t + \delta^k \); these packets enter the scheduler only from \( t - \hat{\tau} \) to \( \min \{ t + \tau, t + \delta^k - \delta^j \} \). Note that the deadlines of packets from session \( j \) that arrive after \( t + \delta^k - \delta^j \) are greater than \( t + \delta^k \). Therefore,

\[
\Pr \left( \left\{ \min_{0 \leq \tau \leq \delta^k + d} \{ W^{k,t} (t + \tau) \} \geq 0 \right\} \cap \{ b(t) = \hat{\tau} \} \right)
\]

\[
= \Pr \left\{ \min_{0 \leq \tau \leq \delta^k + d} \left\{ \sum_{j \in \mathcal{N}} A^j \left[ t - \hat{\tau}, \min \left\{ t + \tau, t + \delta^k - \delta^j \right\} \right] \right\} + R(t - \hat{\tau}) - r(\hat{\tau} + \tau) \right\} \geq 0 \right\},
\]

\[
(4.29)
\]

where \( R(t - b(t)) \) denotes the amount of untransmitted data of a possible packet that is in transmission at time \( t - b(t) \). We distinguish between two cases, namely whether at time \( t - \hat{\tau} \) the scheduler is empty or transmitting a packet.

**Case 1** \( W(t - \hat{\tau}) = 0 \).

In this case, the scheduler is empty at time \( t - \hat{\tau} \), i.e., \( R(t - \hat{\tau}) = 0 \). We obtain from (4.29):

\[
\Pr \left( \left\{ D^k (t) \geq \delta^k + d \right\} \cap \{ b(t) = \hat{\tau} \} \right)
\]

\[
= \Pr \left\{ \min_{0 \leq \tau \leq \delta^k + d} \left\{ \sum_{j \in \mathcal{N}} A^j \left[ t - \hat{\tau}, \min \left\{ t + \tau, t + \delta^k - \delta^j \right\} \right] \right\} - r(\hat{\tau} + \tau) \right\} \geq 0 \right\}
\]

\[
= \Pr \left\{ \min_{0 \leq \tau \leq \delta^k + d} \left\{ \sum_{j \in \mathcal{I}(\tau)} A^j \left[ t - \hat{\tau}, t + \tau \right] + \sum_{j \in \mathcal{N} \setminus \mathcal{I}(\tau)} A^j \left[ t - \hat{\tau}, t + \delta^k - \delta^j \right] - r(\hat{\tau} + \tau) \right\} \geq 0 \right\}
\]

\[
(4.30)
\]
where $j \in \mathcal{I}(\tau)$ if $t + \tau \leq t + \delta^k - \delta^j$. Since the sessions traffic is stationary, it is easy to see that the minimum is achieved for $\mathcal{I}(\tau) = \emptyset$. Thus,

$$
\Pr \left( \left\{ D^k(t) \geq \delta^k + d \right\} \cap \left\{ b(t) = \hat{\tau} \right\} \right) = \Pr \left\{ \sum_{j \in \mathcal{N}} A^j \left[ t - \hat{\tau}, t + \delta^k - \delta^j \right] - r \left( \hat{\tau} + \delta^k + d \right) \geq 0 \right\}.
$$

**Case 2** $W(t - b(t)) > 0$.

The scheduler is transmitting traffic at time $t - \hat{\tau}$ from some connection $j'$. By the definition of $\hat{\tau}$ the traffic in transmission has a deadline greater that $t + \delta^k$, that is, $\delta^j > \hat{\tau} + \delta^k$. Without loss of generality, we assume that $j'$ is such that $L_{j'} = \max_{\delta^j > \hat{\tau} + \delta^k} L_j$. From (4.28) and (4.29) and using similar arguments as in Case 1, we have:

$$
\Pr \left( \left\{ D^k(t) \geq \delta^k + d \right\} \cap \left\{ b(t) = \hat{\tau} \right\} \right) = \Pr \left\{ \min_{0 \leq \tau \leq \delta^k + d} \left\{ \sum_{j \in \mathcal{N}} A^j \left[ t - \hat{\tau}, \min \left\{ t + \tau, t + \delta^k - \delta^j \right\} \right] + \max_{\delta^j > \hat{\tau} + \delta^k} L_j - r \left( \hat{\tau} + \tau \right) \right\} \geq 0 \right\} = \Pr \left\{ \sum_{j \in \mathcal{N}} A^j \left[ t - \hat{\tau}, t + \delta^k - \delta^j \right] + \max_{\delta^j > \hat{\tau} + \delta^k} L_j - r \left( \hat{\tau} + \delta^k + d \right) \geq 0 \right\}.
$$

From (4.28),(4.30) and (4.31) we get

$$
\Pr \left\{ D^k(t) \geq \delta^k + d \right\} \leq \sum_{\hat{\tau} = 0}^t \Pr \left\{ \sum_{j \in \mathcal{N}} A^j \left[ t - \hat{\tau}, t + \delta^k - \delta^j \right] + \max_{\delta^j > \hat{\tau} + \delta^k} L_j \geq r \left( \hat{\tau} + \delta^k + d \right) \right\} \leq q^j.
$$

The proposition follows from the stationarity of the input traffic and by taking $t = t + d^k$ and $\hat{\tau} = \hat{\tau} + d^k$. \hfill \blacksquare

### 4.6.2 Proof of Lemma 4.1

**Lemma 4.1:** Let $\gamma^1 \leq \gamma^2 \leq \ldots \leq \gamma^N$. If the effective delay vector $\left\{ \gamma^1, \gamma^2, \ldots, \gamma^{|\mathcal{N}|} \right\}$ is schedulable under a non-preemptive policy then, for all $j \in \mathcal{N}$ and for all $t$, it holds that

$$
\frac{L_{\max}}{r} \leq \gamma^1
$$

and

$$
\sum_{\hat{\tau} = 0}^t \Pr \left\{ \sum_{j \in \mathcal{N}} A^j \left[ t - \hat{\tau}, t - \gamma^j \right] + \max_{\gamma^j > \hat{\tau}} L_j \geq r \hat{\tau} \right\} \leq q^j
$$

108
Proof. First, assume that all packets meet their deadlines with the required probability under a non-preemptive policy. Clearly, we should have $\frac{L_{\text{max}}}{r} \leq \gamma^1$, since otherwise maximum-length packets from any session are not schedulable. Next, assume that the inequality in (4.34) is violated at time $t > 0$, that is

$$t - \sum_{\hat{\tau} = 0}^{\hat{\tau}} \Pr \left\{ \sum_{j \in N} A^j \left[ t - \hat{\tau}, t - \gamma^j \right] + \max_{\gamma^j \geq \hat{\tau}} L^j \geq r \hat{\tau} \right\} > q^j. \quad (4.35)$$

Now, consider the following scenario. The scheduler is empty at all times up to 0, and at time 0 the last bit of a packet of maximum length from session $|N|$ arrives in the system. At time $0^+$, Exponentially Bounded (EB) bursts from all sessions $i$, $1 \leq i \leq |N|$, of total size $\sigma^i$ each, arrive in the system, where $\Pr \{ \sigma^i \geq x \} \leq \Lambda^i e^{-\alpha^i x}$. Afterwards, packets from all sessions $i$, $1 \leq i \leq |N|$, arrive at fixed rate $\rho^i$. Accordingly, we have that $A^j [0, t - \gamma^j] = \sigma^j + \rho^j (t - \gamma^j)$ and that for all $\hat{\tau} < t$: $A^j [t - \hat{\tau}, t - \gamma^j] = \rho^j (\hat{\tau} - \gamma^j)$. By (4.33) and the stability condition ($\sum_{j \in N} \rho^j < r$) we have that

$$\Pr \left\{ \sum_{j \in N} \rho^j (\hat{\tau} - \gamma^j) + L_{\text{max}} \geq r \hat{\tau} \right\} = 0.$$

Thus,

$$\sum_{\hat{\tau} = 0}^{t-} \Pr \left\{ \sum_{j \in N} A^j \left[ t - \hat{\tau}, t - \gamma^j \right] + \max_{\gamma^j \geq \hat{\tau}} L^j \geq r \hat{\tau} \right\} = 0.$$

With the assumption from (4.35) we have

$$\sum_{\hat{\tau} = 0}^{t-} \Pr \left\{ \sum_{j \in N} A^j \left[ t - \hat{\tau}, t - \gamma^j \right] + \max_{\gamma^j \geq \hat{\tau}} L^j \geq r \hat{\tau} \right\} + \Pr \left\{ \sum_{j \in N} A^j \left[ 0, t - \gamma^j \right] + \max_{\gamma^j \geq \hat{\tau}} L^j \geq r t \right\} > q^j,$$

consequently,

$$\Pr \left\{ \sum_{j \in N} A^j \left[ 0, t - \gamma^j \right] + \max_{\gamma^j \geq \hat{\tau}} L^j \geq r t \right\} > q^j. \quad (4.36)$$

The workload in the scheduler that should be transmitted by time $t$ is

$$\sum_{j \in N} A^j \left[ 0, t - \gamma^j \right] + \max_{\gamma^j \geq \hat{\tau}} L^j = \sum_{j \in N} (\sigma^j + \rho^j (t - \gamma^j)) + \max_{\gamma^j \geq \hat{\tau}} L^j.$$

Therefore, the probability that there is a packet in the scheduler at time $t$ (say from session $j$) with delay violation is greater than $q^j$. \qed
Proposition 4.4: Let
\[
D_i(A_k, A_{k+1}) (s_i^k) = \frac{1}{\rho} \left[ W_\rho (A_k) (s_i^k) - [\beta_{k+1} W_\rho (A_0) (s_i) - \varepsilon_{k+1}]^+ \right]. \tag{4.37}
\]
For all \( k = 1, 2, \ldots, n \) it holds that:
1. \( W_\rho (A_k) (t_i^k) = W_\rho (A_0) (s_i) - \rho D_i(A_0, A_{1:k}) \)
2. \( W_\rho (A_n) (t_i^n) \leq [\beta_k W_\rho (A_0) (s_i) - \varepsilon_k]^+ \)
3. for all \( \sigma > 0, s > t > 0, \) \( \Pr \{ A_n (t, s) \geq \rho (s - t) + \sigma \} \leq \Lambda_k e^{-\alpha_k \sigma}, \)
   where \( \Lambda_k = \frac{\Lambda_0}{1 - e^{-\alpha_0 \rho \sigma}} e^{-\alpha_k \sigma} \) and \( \alpha_k = \frac{\alpha m}{\beta k} \).
4. \( \Pr \{ D_i(A_0, A_{1:k}) \geq d \} \leq \max_{m=1,2,\ldots,k} \frac{\Lambda_0}{1 - e^{-\alpha_0 \rho \sigma}} e^{\alpha_0 \frac{e m}{1 - \beta m}} e^{-\alpha_0 \frac{e m}{1 - \beta m}}. \)

**Proof.** From the proof of proposition 4.3 we have that
\[
W_\rho (A_1) (t_i) = W_\rho (A_0) (s_i) - \rho D_i(A_0, A_1). \tag{4.38}
\]
To prove the first part of the proposition, we apply (4.38) \( n \) times: in the \( k \)-th application, we use \( A_{k-1}, A_k, s_i + D_i(A_0, A_{1:k-1}), s_i + D_i(A_0, A_{1:k}), D_i(A_0, A_{1:k}) - D_i(A_0, A_{1:k-1}), \rho \) instead of \( A_0, A_1, s_i, t_i, d_i, \rho \), respectively. This implies:
\[
W_\rho (A_k) (t_i^k) = W_\rho (A_0) (s_i) - \rho D_i(A_0, A_{1:k}), \tag{4.39}
\]
for all \( k = 1, 2, \ldots, n. \)

In particular, it follows that \( W_\rho (A_m) (t_i^m) \) is nonincreasing in \( m \) for each fixed \( i \). On the other hand, due to the \( k \)-th traffic shaper we have
\[
W_\rho (A_k) (t_i^k) \leq (\beta_k W_\rho (A_0) (s_i) - \varepsilon_k)^+. \]
Thus the second part follows.

The third part of the proposition follows from the second part and from proposition 4.3.

To prove the third part of the proposition we establish by induction on \( k \) that
\[
D_i(A_0, A_{1:k}) = \max_{m=1,2,\ldots,k} D_i(A_0, A_m) \tag{4.40}
\]
It holds for \( k = 1 \) by the definition of \( D_i(A_0, A_{1:k}) \). Assuming that it holds for some \( k \), we now show that it holds for \( k + 1 \). From (4.37) and (4.39) we have
\[
D_i(A_0, A_{1:k+1}) - D_i(A_0, A_{1:k}) =
\[
\frac{1}{\rho} \left[ W_\rho (A_0) (s_i) - \rho D_i(A_0, A_{1:k}) - [\beta_{k+1} W_\rho (A_0) (s_i) - \varepsilon_{k+1}]^+ \right]. \tag{4.41}
\]
If \( D_i(A_0, A_{1:k+1}) - D_i(A_0, A_{1:k}) > 0 \) then the second `+` superscript in (4.41) can be removed, yielding

\[
D_i(A_0, A_{1:k+1}) = \frac{1}{\rho} \left( W_\rho(A_0)(s_i) - [\beta_{k+1} W_\rho(A_0)(s_i) - \varepsilon_{k+1}]^+ \right). 
\]

On the other hand, if \( D_i(A_0, A_{1:k+1}) - D_i(A_0, A_{1:k}) = 0 \) it follows from (4.41) that

\[
\frac{1}{\rho} \left( W_\rho(A_0)(s_i) - [\beta_{k+1} W_\rho(A_0)(s_i) - \varepsilon_{k+1}]^+ \right) \leq D_i(A_0, A_{1:k}). 
\]

Hence,

\[
D_i(A_0, A_{1:k+1}) = \max \left\{ D_i(A_0, A_{1:k}), \frac{1}{\rho} \left( W_\rho(A_0)(s_i) - [\beta_{k+1} W_\rho(A_0)(s_i) - \varepsilon_{k+1}]^+ \right) \right\}. 
\]

This completes the induction step.

The last part of the proposition follows from Proposition 4.3, (4.40) and the following relation:

\[
\Pr \left\{ \max_{m=1,2,...,k} D_i(A_0, A_m) \geq d \right\} \leq \max_{m=1,2,...,k} \Pr \{ D_i(A_0, A_m) > d \}. 
\]

### 4.6.4 Proof of Lemma 4.2

**Lemma 4.2:** Assume that packets arrive to systems \( S_1, S_2 \) according to the same arrival process \( A(t) \). If \( D_i^{(1)} \) and \( D_i^{(2)} \) are the delays of packet \( i \) in the traffic shaper in \( S_1 \) and \( S_2 \) respectively, then, for all \( i = 1, 2, \ldots, \),

\[
D_i^{(1)} \leq D_i^{(2)} + \theta_i \tag{4.42} 
\]

**Proof.** The proof of the Lemma goes along similar lines to that of Lemma 1 in [4] and is specified for completeness.

Let \( s_i^{(1)} \) be the time the \( i \)-th packet of length \( L_i \) that arrives to each of the two systems. Also, let \( s_i^{(2)} = s_i^{(1)} + \theta_i \) be the time that packet \( i \) enters the shaper in \( S_2 \). According to (4.12) and (4.13) we have, for \( n = 1, 2, \ldots \),

\[
D_i^{(n)} = \frac{1}{\rho} \left( W_\rho(A^{(n)})(s_i^{(n)}) - [\beta W_\rho(A_0)(s_i) - \varepsilon]^+ \right) 
\]

\[
= \max_{0 \leq s \leq s_i^{(n)}} \left\{ A^{(n)}[s, s_i^{(n)}] - \rho \left( s_i^{(n)} - s \right) - \sigma_i \right\}, \tag{4.43} 
\]

111
where

\[ \sigma_i = [\beta W_\rho (A_0) (s_i) - \varepsilon]^+. \]

We have that

\[ A^{(n)} \left[ s, s^{(n)}_{i+1} \right] = \begin{cases} A^{(n)} \left[ s, s^{(n)}_i \right] + L_{i+1} & s \leq s^{(n)}_i \\ L_{i+1} & s^{(n)}_i < s \leq s^{(n)}_{i+1} \end{cases}. \]

Accordingly, the quantities \( D^{(n)}_{i+1}, n = 1, 2 \) satisfy,

\[ D^{(n)}_{i+1} = \max \left\{ \max_{0 \leq s \leq s^{(n)}_i} \left\{ \frac{A^{(n)} \left[ s, s^{(n)}_i \right] - \rho \left( s^{(n)}_i - s \right) - \sigma_i + L_{i+1}}{\rho} - \left( \frac{s^{(n)}_i - s^{(n)}_{i+1}}{\rho} \right) \left( \sigma_{i+1} - \sigma_i \right) \right\}, \right. \]

\[ \left. \max_{s^{(n)}_i < s \leq s^{(n)}_{i+1}} \left\{ \frac{L_{i+1} - \sigma_{i+1}}{\rho} - \left( \frac{s^{(n)}_i - s^{(n)}_{i+1}}{\rho} \right) \left( \sigma_{i+1} - \sigma_i \right) \right\} \right\}, \]

\[ = \max \left\{ D^{(n)}_i + \frac{L_{i+1}}{\rho} - \left( \frac{s^{(n)}_i - s^{(n)}_{i+1}}{\rho} \right) - \left( \sigma_{i+1} - \sigma_i \right), L_{i+1} - \sigma_{i+1} \right\}. \] (4.44)

We shall show by induction that for all \( i = 1, 2, \ldots \)

\[ D^{(1)}_i \leq D^{(2)}_i + \theta_i, \] (4.45)

which proves the Lemma. For \( i = 1 \) we have from (4.43) that

\[ D^{(1)}_1 = \frac{L_1 - \sigma_1}{\rho} = D^{(2)}_1 \leq D^{(2)}_1 + \theta_1. \]

Assuming that (4.45) holds for \( i \), we then have from (4.44),

\[ D^{(2)}_{i+1} + \theta_{i+1} \]

\[ = \max \left\{ D^{(2)}_i + \frac{L_{i+1}}{\rho} - \left( \frac{s^{(2)}_{i+1} - s^{(2)}_i}{\rho} \right) - \left( \sigma_{i+1} - \sigma_i \right), L_{i+1} - \sigma_{i+1} \right\} + \theta_{i+1} \]

\[ = \max \left\{ D^{(2)}_i + \frac{L_{i+1}}{\rho} + \left( s^{(1)}_{i+1} - s^{(1)}_i \right) - \left( \sigma_{i+1} - \sigma_i \right), L_{i+1} - \sigma_{i+1} + \theta_{i+1} \right\} + \theta_{i+1} \]

\[ = \max \left\{ D^{(2)}_i + \theta_i + \frac{L_{i+1}}{\rho} - \left( \frac{s^{(1)}_{i+1} - s^{(1)}_i}{\rho} \right) - \left( \sigma_{i+1} - \sigma_i \right), L_{i+1} - \sigma_{i+1} + \theta_{i+1} \right\} \]

\[ \geq \max \left\{ D^{(1)}_i + \frac{L_{i+1}}{\rho} - \left( \frac{s^{(1)}_{i+1} - s^{(1)}_i}{\rho} \right) - \left( \sigma_{i+1} - \sigma_i \right), L_{i+1} - \sigma_{i+1} \right\} \]

\[ = D^{(1)}_{i+1}. \]
Chapter 5

Call Admission Control in Fault-Tolerant Networks

5.1 Introduction

Over the years, data networks have undergone significant changes not only with respect to the volume and nature of the traffic they carry, but also with respect to the criticality of their role and the level of service demanded from them. In particular, data networks are expected to have fault tolerance and restoration capability, i.e., ability to reconfigure and re-establish original communication upon failures.

A network with restoration capability requires redundant capacity to be used in case of failures. Upon a resource failure, the connection is re-connected through a backup resource. If there are no backup resources, the connection is terminated. Such a premature disconnection is highly undesirable and one of the goals of a fault tolerant network designer is to keep the probability of such occurrences low. On the other hand, reserving resources for restoration could increase blocking of new connections. As a result, there is a trade-off between the two Quality-of-Service (QoS) measures, namely, restoration capability and blocking probabilities of incoming calls. In this study, we consider admission control policies for three network scenarios based on these two QoS measures. More specifically, we first consider optical networks with shared protection; then, we turn to consider wireless networks.

The call admission control problem has been extensively studied under the context of Markov Decision Process (MDP) (see, e.g., [46, 47, 48] and references therein). In particular, the authors of [48] considered call admission control of multiple classes without waiting room. Using event-based dynamic programming, as introduced in [49], they showed that, under certain conditions the customer classes can be ordered in the following way: if it is optimal to accept a class, then to accept a more profitable class is optimal too. Furthermore, they established the optimality of a trunk reservation policy for a special case with two classes of customers, referred as the one-dimensional case. A similar model, with two classes of traffic and in the
context of cellular communication, has been studied in [50].

Optical networks, with their terabit-wide links, are emerging as the predominant transport layer technology, and are chartered to carry high volumes of traffic. Yet, Optical networks are prone to component failures. A fiber cut causes a link failure. A node failure may be caused due to the failure of the associated Optical Cross Connect (OXC). A fiber may fail due to the failure of its connected end-components. Since optical networks carry high volumes of traffic loads, failures may result in severe loss. Therefore, it is imperative that these networks have high fault tolerance capabilities. Indeed, lightpath protection and restoration in optical networks have been the subject of several studies and proposals (see, e.g., [51, 52, 53, 54] and references therein).

We consider a network of \( M \) lightpaths connecting two end nodes. We assume that each connection is transmitted through a lightpath. In case of a lightpath failure, the connection that was transmitted in that lightpath is re-routed to a different lightpath, if available. If there is no available lightpath, then the connection is terminated and a penalty is incurred. On the other hand, a connection request that is admitted to the system gives rise to a reward. Under this setting, we consider the admission control policy that maximizes the total discounted reward, and establish the optimality of the trunk reservation policy. More specifically, we show that the optimal policy is of a threshold type, i.e., for each number of operating lightpaths (out of \( M \)) there is a critical level above which no connections are established.

Wireless local area networks (WLANs) have increasingly become the edge network of choice. Hot spots are widely deployed in diverse places such as coffee shops, airports, hotels and shopping centers. Concurrent with the expansion of WLANs is a high demand for quality of service (QoS)-sensitive applications for a variety of professional and personal uses. For example, WLANs are being used in residential networks to support a wide range of applications such as remote controls, video from a security camera, delivery of video on demand, voice telephony, streaming audio, and Internet access. It has also been suggested to carry WLAN to home users over the Cable TV infrastructure, in order to provide wireless WAN internet access [55].

However, as is the case with wireless networks in general, lack of bandwidth and interference constraints make WLANs a potential bottleneck. Furthermore, the amount of wireless bandwidth is subject to the channel conditions. Specifically, the IEEE 802.11b supports payload data rates of 1, 2, 5.5 and 11 Mbit/s and the IEEE 802.11g provide additional rates of 6, 9, 12, 18, 24, 36, 48 and 54 Mbit/s. Of these rates, transmission rate is determined according to the quality of the channel. That is, upon high rate of packet loss, the transmitting station decreases its data rate.

Accordingly, upon decrease of the transmission rate, wireless connections might be terminated due to lack of bandwidth. In the context of WLANs, fault toler-
ance refers to the ability to serve all connections at the event of rate decrease. Since the WLAN MAC mechanism is typically (e.g., in IEEE 802.11) based on the CSMA/CA protocol, it is essential to maintain the offered load smaller than the maximal throughput, otherwise the system could be unstable, i.e., the throughput would decrease with the offered load (due to retransmissions). Thus, blocking connections in order to maintain stability could be imperative.

Clearly, the termination of a connection is highly undesirable. On the other hand, reserving bandwidth, to account for possible bad channel conditions, increases the blocking probability of new connections. Again, there is a tradeoff between premature termination of connections, due to a decrease in the transmission rate, and the blocking probabilities on incoming calls. Under certain assumptions on the channel and the connection characteristics, we establish the optimality of a trunk reservation policy.

5.2 Optical networks with shared protection

Given are $N$ resources (lightpaths) in parallel. We shall henceforth use the terms "lightpath" and "resource" interchangeably. Connection requests arrive according to a Poisson process with parameter $\lambda$ and the service time is exponentially distributed with parameter $\mu$. Each connection requires a single resource. Resources occasionally fail, with mean time between failures that is exponentially distributed with parameter $\sigma$. Repair time is exponentially distributed as well, with parameter $\tau$. In case of a resource failure, the connection that was served by that resource is rerouted to a different lightpath, if available. If there is no available lightpath, then the connection is terminated and a penalty ($-p$) is incurred. When a connection request arrives to the system, a central controller rejects or accepts the connection based on full state information. When a connection is accepted, a reward $r$ is incurred. We assume that $p > r$; otherwise, it follows immediately that the optimal policy would accept all connection requests as long as there are available resources.

Consider the following admission control problem: find the optimal strategy (decision whether to accept or reject a call) such that the total discounted reward is maximized.

Let $x$ be the number of connections and $m$ the number of available resources, where $x \leq m \leq M$ and $M$ is the total number of available resources. The state of the system is given by $(x, m)$. A symbolic representation of the state transition structure for the above admission control model is depicted in Figure 5.1. The figure illustrates the transitions at state $(x, m)$, where $0 < x < m < M$. Let $q_d(j|s) = q(j|s, d(s))$ denote the probability that at the subsequent decision epoch the system occupies state $j$, if at the current decision epoch it occupies
state $s$ and the decision rule is $d(s)$. Furthermore, the process remains in state $s$ for a period of time determined by an exponential distribution with parameter $b_d(s) = b(s, d(s))$, $0 \leq b_d(s) < \infty$. Then, the values (on the arrows) in Figure 5.1 correspond to $b_d(s) \cdot q_d(j|s)$. Consider, for example, an arrival event at state $(x, m)$ and assume that $d((x, m)) = 1$ (i.e., the connection is admitted). We have that, $b_d((x, m)) = \lambda + x\mu + m\sigma + (M - m)\tau$, thus, $b_d((x, m)) q_{d=1}((x + 1, m) | (x, m)) = \lambda$.

![Figure 5.1: Symbolic representation of the state transition structure for the admission control model](image)

The standard approach to solve the above admission control problem is to employ uniformization [46] and formulate the value function as:

$$\tilde{T}v = \max_{d \in D^MDP} \left\{ \tilde{r}_d(s) + \tilde{\beta} \sum_{j \in S} \tilde{q}_d(j|s) v(j) \right\},$$

(5.1)

where, $D^{MDP}$ is the set of Markov deterministic decision rules, $\tilde{r}_d(s) = \tilde{r}(s, d(s))$ is the expected reward/cost after uniformization, $\tilde{\beta}$ is the discount factor after uniformization and $\tilde{q}_d(j|s) = \tilde{q}(j|s, d(s))$ is the transition probability after uniformization. Figure 5.2 illustrates the transitions structure for the uniformization of the admission control problem at state $(x, m)$, where $0 < x < m < M$. Arcs denoted by the ··· have been altered from the original model. From the theory of continuous-time Markov Decision Processes (see, e.g., [56]), under certain assumptions, if the maximum in (5.1) exists for all $v \in V$ (where $V$ is the space of bounded functions on $S$), there exists a stationary deterministic optimal policy, which can be found by the optimality equation $v = \tilde{T}v$. However, showing the optimality of a threshold policy
directly from the optimality equation is not trivial. Thus, following [48], we solve this problem by inductively establishing properties of the dynamic programming value function. To formulate the dynamic programming equation, we shall employ event-based dynamic programming, as introduced in [49].

Specifically, to every possible event in the system there corresponds a dynamic programming event operator that maps the set $F$ of all real-valued functions of the state variable $(x, m)$ into itself.

For connection request arrivals, the corresponding arrival operator $T_A$ is defined by

$$T_A f(x, m) = \max \{ r + f(x + 1, m), f(x, m) \},$$

for $f \in F$. The function $T_A f(x, m)$ may be interpreted as the optimal value function for a one-stage problem in which one needs to decide whether to accept or reject a connection request, after which a revenue is incurred according to the function $f$. If the connection is established, then a reward $r$ is incurred and the state changes to $(x + 1, m)$, otherwise the state remains unchanged, i.e., $(x, m)$.

Similarly, for a connection “departure” from the multi-server system with $M$ servers, we have (see Figure 5.2)

$$T_{MD} f(x, m) = \begin{cases} xf(x - 1, m) + (M - x) f(x, m) & x < M \\ M f(x - 1, m) & \text{otherwise} \end{cases}.$$

Next, $T_{MF}$ corresponds to failures of light paths. If a lightpath fails and there is no other resource available, i.e., $x = m$, then a connection is terminated and a penalty
of $-p$ is imposed. Therefore:

$$T_{MF}f(x,m) = \begin{cases} 
mf(x,m-1) + (M-m) f(x,m) & x < m \\
m(f(x-1,m-1) - p) + (M-m) f(x,m) & x = m \end{cases}.$$ 

For a lightpath repair event, we have

$$T_{MR}f(x,m) = (M-m) f(x,m+1) + mf(x,m).$$

Finally, let $C(x,m)$ denote the costs associated with a state $(x,m)$. These costs are only used to prevent the system from leaving the state space, thus we set:

$$C(x,m) = \begin{cases} 
0 & x \leq m \leq M \\
\infty & \text{otherwise} 
\end{cases}.$$

Using the above event operators and the uniformization technique [46], we have

$$T f(x,m) = -C(x,m) + \beta [\lambda T_A f(x,m) + \mu T_{MD} f(x,m) + \sigma T_{MF} f(x,m) + \tau T_{MR} f(x,m)].$$

Without loss of generality, we have assumed that $\lambda + M\mu + M\sigma + M\tau = 1$; $\beta \in (0,1)$ is the discounted factor.

In the following, we prove certain properties of the value function $V_n$, defined by $V_{n+1}(x,m) = TV_n(x,m)$ and $V_0(x,m) = -C(x,m)$. As these properties hold for all $n$, they hold also for the limiting optimal policy, as follows from standard results in Markov Decision theory (see e.g. [56]).

First, we show that $V_n$ is non-increasing in $x$.

**Lemma 5.1** For all $x$, $m$, and $n$, we have that

$$V_n(x,m) \geq V_n(x+1,m). \quad (5.2)$$

**Proof.** It is easily seen that $V_0$ (i.e., $-C$) satisfies the above inequality. We show that, if $f(x,m)$ satisfies (5.2), then so does $T \circ f(x,m)$ for all event operators $T \circ$, $T \circ \in \{T_A, T_{MD}, T_{MF}, T_{MR}\}$. As the inequality is closed under linear combinations, the lemma follows directly by induction on $n$.

**$T_A$:** Suppose that $f(x,m)$ is non-increasing in $x$. Then, for all $x$ and $m$

$$T_A f(x,m) = \max \{ r + f(x+1,m), f(x,m) \} \geq \max \{ r + f(x+2,m), f(x+1,m) \} = T_A f(x+1,m).$$

This shows that $T_A f(x,m)$ is non-increasing in $x$. 

118
**TMD:**

\[ T_{MD} f(x, m) = xf(x - 1, m) + (M - x) f(x, m) \]
\[ \geq (x + 1) f(x, m) + (M - x - 1) f(x + 1, m) \]
\[ = T_{MD} f(x + 1, m). \]

**TMF:** If \( x + 1 < m \) then

\[ T_{MF} f(x, m) = mf(x, m - 1) + (M - m) f(x, m) \]
\[ \geq m (f(x, m - 1) - p) + (M - m) f(x + 1, m) \]
\[ = T_{MF} f(x + 1, m), \]

otherwise, if \( x + 1 = m \) then

\[ T_{MF} f(x, m) = mf(x, m - 1) + (M - m) f(x, m) \]
\[ \geq m (f(x, m - 1) - p) + (M - m) f(x + 1, m) \]
\[ = T_{MF} f(x + 1, m). \]

**TMR:**

\[ T_{MR} f(x, m) = (M - m) f(x, m + 1) + mf(x, m) \]
\[ \geq (M - m) f(x + 1, m + 1) + mf(x + 1, m) \]
\[ = T_{MR} f(x + 1, m). \]

Recall that \( \lambda + M\mu + M\sigma + M\tau = 1 \) and \( \beta < 1 \). Thus, by the above inequalities, we have that \( T f(x, m) \geq T f(x + 1, m). \) 

The following technical lemma is required for the proof of the next Theorem.

**Lemma 5.2** For all \( 0 \leq x \leq m - 1, 0 \leq m \leq M \) and \( n \), we have that

\[ V_n(x + 1, m) \geq V_n(x, m) - p. \]  

(5.3)

**Proof.** It is easily seen that \( V_0 \) (i.e., \(-C\)) satisfies the above inequality for all \( 0 \leq x \leq m - 1 \). We show that, if \( f(x, m) \) satisfies inequality (5.3), then so does \( T\_\bullet f(x, m) \) for all event operators \( T\_\bullet \). The lemma follows directly by induction on \( n \).

**T\_A:** Suppose that \( f(x, m) \) satisfies (5.3). Then, for all \( 0 \leq x < m - 1 \) and \( 0 \leq m \leq M \)

\[ T\_A f(x + 1, m) = \max \{ r + f(x + 2, m), f(x + 1, m) \} \]
\[ \geq \max \{ r + f(x + 1, m), f(x, m) \} - p \]
\[ = T\_A f(x, m) - p. \]
Now, considering $x + 1 = m$, there are no available resources for new connections, thus, we have that

$$T_A f(x + 1, m) = f(x + 1, m) \geq \max \{ r + f(x + 1, m), f(x, m) \} - p = T_A f(x, m) - p,$$

where the inequality follows from the inductive assumption (i.e. inequality (5.3)) and since $p > r$.

**$T_{MD}$**: It follows from the inductive assumption that

$$T_{MD} f(x + 1, m) = (x + 1) f(x, m) + (M - x - 1) f(x + 1, m) \geq x f(x - 1, m) - xp + f(x, m) + (M - x) f(x, m) - (M - x) p - f(x + 1, m) \geq T_{MD} f(x, m) - Mp.$$  

**$T_{MF}$**: It immediately follows that, for $x + 1 < m$:

$$T_{MF} f(x + 1, m) = mf(x + 1, m - 1) + (M - m) f(x + 1, m) \geq mf(x, m - 1) - mp + (M - m) f(x, m) - (M - m) p \geq T_{MF} f(x, m) - Mp.$$  

Now, for $x + 1 = m$, we have:

$$T_{MF} f(x + 1, m) = mf(x, m - 1) - mp + (M - m) f(x + 1, m) \geq mf(x, m - 1) - mp + (M - m) f(x, m) - (M - m) p \geq T_{MF} f(x, m) - Mp.$$  

**$T_{MR}$**:

$$T_{MR} f(x + 1, m) = (M - m) f(x + 1, m + 1) + mf(x + 1, m) \geq (M - m) f(x, m + 1) - (M - m) p + mf(x, m) - mp \geq T_{MR} f(x, m) - Mp.$$  

Recall that

$$T f(x, m) = -C(x, m) + \beta [\lambda T_A f(x, m) + \mu T_{MD} f(x, m) + \sigma T_{MF} f(x, m) + \tau T_{MR} f(x, m)],$$
\[ \lambda + M\mu + M\sigma + M\tau = 1 \] and \( \beta < 1 \). Thus, by the above inequalities, we have that \( T_f (x + 1, m) \geq T_f (x, m) - p \).

The following theorem shows that the optimal call admission control policy uses the idea of trunk reservation, i.e., for each number of available resources there is a threshold above which no connections are admitted. The thresholds assure that some of the resources are kept free (reserved) for restoration. Those thresholds can be calculated numerically, e.g., by the value iteration procedure (see e.g., [56]).

**Theorem 5.1**

1. For all \( 0 \leq x \leq m - 2, 0 \leq m \leq M \) and \( n \), we have that

\[
2V_n (x + 1, m) \geq V_n (x, m) + V_n (x + 2, m),
\]

i.e., \( V_n \) is concave with \( x \).

2. The optimal policy is of the threshold form, i.e., for each number of available resources there is a critical level above which no connection is admitted.

**Proof.** It is easily seen that \( V_0 \) (i.e. \(-C\)) satisfies (5.4). We show that, if \( f (x, m) \) satisfies (5.4), then so does \( T_a f (x, m) \) for all event operators \( T_a \), \( T_a \in \{T_A, T_{MD}, T_{MF}, T_{MR}\} \). As the inequality is closed under linear combinations, the lemma follows directly by induction on \( n \).

Consider the operator \( T_A \). Denote by \( a_1 \) (correspondingly, \( a_2 \)) the maximizing action in \( T_A f (x, m) \) (correspondingly, \( T_A f (x + 2, m) \)), where an action 0 (correspondingly, 1) refers to rejecting (correspondingly, accepting) a connection. Consider the following cases:

if \( a_1 = a_2 = 1 \), then

\[
2T_A f (x + 1, m) = 2 \max \{ r + f (x + 2, m), f (x + 1, m) \}
\geq 2r + 2f (x + 2, m).
\]

if \( a_1 = a_2 = 0 \), then

\[
2T_A f (x + 1, m) = 2 \max \{ r + f (x + 2, m), f (x + m) \}
\geq 2f (x + 1, m).
\]

\[
= T_A f (x, m) + T_A f (x + 2, m);
\]
if \(a_1 = 1\) and \(a_2 = 0\), then
\[
2T_A f(x + 1, m) = 2 \max \{r + f(x + 2, m), f(x + 1, m)\} \\
\geq r + f(x + 2, m) + f(x + 1, m) \\
= T_A f(x, m) + T_A f(x + 2, m);
\]
if \(a_1 = 0\) and \(a_2 = 1\), then
\[
2T_A f(x + 1, m) = 2 \max \{r + f(x + 2, m), f(x + 1, m)\} \\
\geq r + f(x + 2, m) + f(x, m) + f(x + 2, m) - f(x + 1, m) \\
\geq f(x, m) + r + f(x + 3, m) \\
= T_A f(x, m) + T_A f(x + 2, m);
\]
and the concavity of \(T_A\) with respect to \(x\) follows.

Consider next \(T_{MF}\). The result follows easily for \(x + 2 < m\). Consider, then, \(x + 2 = m\). We have:
\[
2T_{MF} f(x + 1, m) \\
= 2mf(x + 1, m - 1) + 2(M - m) f(x + 1, m) \\
\geq 2mf(x + 1, m - 1) + (M - m) f(x, m) + (M - m) f(x + 2, m) \\
\geq mf(x, m - 1) + [mf(x + 1, m - 1) - mf(x, m - 1)] + mf(x + 1, m - 1) \\
\quad + (M - m) f(x, m) + (M - m) f(x + 2, m) \\
\geq mf(x, m - 1) - mp + mf(x + 1, m - 1) \\
\quad + (M - m) f(x, m) + (M - m) f(x + 2, m) \\
= mf(x, m - 1) + (M - m) f(x, m) \\
\quad + m(-p + f(x + 1, m - 1)) + (M - m) f(x + 2, m) \\
= T_{MF} f(x, m) + T_{MF} f(x + 2, m).
\]
The last inequality is derived from Lemma 5.2.

The terms concerning \(T_{MD}\) and \(T_{MR}\) are straightforward and the first part of the theorem follows.

By the first part of the theorem, we have that, for any \(m \leq M\),
\[
r + V_n(x + 1, m) - V_n(x, m) \geq r + V_n(x + 2, m) - V_n(x + 1, m).
\]
Thus if \(r + V_n(x + 2, m) - V_n(x + 1, m) \geq 0\), i.e., admission is optimal in state \((x + 1, m)\),
then also \( r + V_n(x + 1, m) - V_n(x, m) \geq 0 \), i.e., admission is optimal in state \((x, m)\). Similarly, we can show that, if rejection is optimal in \((x, m)\), then so it is in state \((x + 1, m)\).

5.3 Wireless LAN networks

We consider a WLAN environment, in which connections from wireless stations share a wireless media. We make the following assumptions, connection requests arrive according to a Poisson process, with parameter \( \lambda \). All connections demand the same bandwidth \( b \) (which corresponds to, e.g., the effective bandwidth of the connections). Such a scenario is appropriate for a WLAN system with connections of similar characteristics; for example, an access point that provides access to WiFi telephony applications. Without loss of generality, we assume that \( b = 1 \). The bandwidth consumed by a connection is released after an exponentially distributed service time, with parameter \( \mu \).

Markov processes with finite number of states are widely used to model wireless channels between two stations (e.g., [57, 58]). This model originates from the ”Gilbert-Elliot model” [59, 60] with the two states named Good and Bad. We extend this model to a Markov chain with \( M \) states, where each channel state is associated with a different transmission rate.

Accordingly, we assume that the bandwidth of the wireless media takes one out of \( M \) possible values, according to a Markov chain. That is, the bandwidth occasionally decreases, with mean time between ”failures” that is exponentially distributed with parameter \( \sigma \). Similarly, the bandwidth increases within an exponentially distributed time, with parameter \( \tau \). Furthermore, we assume that all stations transmit at the same rate, i.e., the wireless channels between all stations are at the same state at all times.

The system states are denoted by \((x, m)\), where \( x \) signifies the number of connections in the system and \( m \) signifies the state of the channel. The system available bandwidth is denoted by \( B(m) \), where \( m = 1, 2, \ldots, M \) and \( B(1) \leq B(2) \leq \ldots \leq B(M) \). Let \( B^m = \left\lfloor \frac{B(m)}{b} \right\rfloor = \lfloor B(m) \rfloor \). We assume that, at state \((x, m)\), it holds that \( x \leq y(m) \), i.e., there is no “waiting room”, and connections that do not find sufficient bandwidth are automatically blocked. There is a central controller (access point) that can reject arriving connections based on full state information. Each connection that enters the system gives rise to a reward of \( r \). However, in case that the channel state deteriorates to \( m - 1 \) and the consumed bandwidth exceed \( B^{m-1} \), then some connections are terminated. In that case, for each connection that is terminated, the system incurs a penalty of \(-p\). We assume that \( p > r \). We shall study admission policies for this model for the discounted
reward criterion.

The WLAN model considered in this section differs from the optical network model in Section 5.2 in the following factors. First, each optical connection consumes one resource whereas wireless connections consumes a share of the wireless bandwidth. Furthermore, at the occurrence of a lightpath failure only one connection might be terminated, whereas a bandwidth decrease in wireless networks might lead to the termination of several connections. Second, and more important, the optical media is composed of several resources that fail (and repaired) independently, whereas the wireless media is modeled by a single resource with varying bandwidth.

To every possible event in the system there corresponds a dynamic programming event operator that maps the set $F$ of all real-valued functions of the state variable $(x, m)$ into itself.

For arrivals of a connection request, we have that the corresponding arrival operator $T_A$ is defined by

$$T_A f(x, m) = \max \{ r + f(x + 1, m), f(x, m) \},$$

for $f \in F$.

For a connection “departure” from the system, we have:

$$T_{MD} f(x, m) = \begin{cases} x f(x, m) + (B^m - x) f(x, m) & x < B^m \\ B^m f(x, m) & \text{otherwise} \end{cases}.$$

Next, $T_F$ corresponds to channel ”failures”. If the channel bandwidth decreases to $B^{m-1}$ and there is not sufficient bandwidth, i.e., $x > B^{m-1}$, then a failure penalty of $-p(x - B^{m-1})$ is incurred. Thus:

$$T_F f(x, m) = \begin{cases} f(x, m - 1) & x \leq B^{m-1} \\ f(B^{m-1}, m - 1) - p(x) & x > B^{m-1} \end{cases}.$$

For a ”repair” event, we have:

$$T_R f(x, m) = \begin{cases} f(x, m + 1) & 1 \leq m < M \\ f(x, m) & \text{otherwise} \end{cases}.$$

Finally, let $C(x, m)$ denote the costs associated with a state $(x, m)$. These costs are only used to prevent the system from leaving the state space, thus we set:

$$C(x, m) = \begin{cases} 0 & 0 \leq x \leq B^m, 1 \leq m \leq M \\ \infty & \text{otherwise} \end{cases}.$$

Using the above event operators and the uniformization technique [46], we have

$$T f(x, m) = -C(x, m) + \beta [\lambda T_A f(x, m) + \mu T_{MD} f(x, m) + \sigma T_F f(x, m) + \tau T_R f(x, m)].$$
We have assumed, without the loss of generality that \( \lambda + B^M \mu + \sigma + \tau = 1; \beta \in (0,1) \) is the discounted factor.

First, we show that \( V_n \) is non-increasing in \( x \).

**Lemma 5.3** For all \( x, m, \) and \( n \), we have that

\[
V_n(x, m) \geq V_n(x + 1, m). \tag{5.5}
\]

**Proof.** The lemma follows by induction on \( n \). It is easily seen that \( V_0 \) (i.e., \( -C \)) satisfies the above inequality. We show that, if \( f(x, m) \) satisfies (5.5), then so does \( T_* f(x, m) \) for all event operators \( T_*, T_0 \in \{ T_A, T_{MD}, T_F, T_R \} \). The proof for the operators \( T_A \) and \( T_{MD} \) is similar to that in Lemma 5.1. We turn to consider the operators \( T_F \) and \( T_R \).

**\( T_F \):** If \( x + 1 \leq B^{m-1} \) then

\[
T_F f(x, m) = f(x, m - 1) \\
\geq f(x + 1, m - 1) \\
= T_F f(x + 1, m);
\]

otherwise, if \( x + 1 > B^{m-1} \) then

\[
T_F f(x, m) = f(B^{m-1}, m - 1) - p(x - B^{m-1}) \\
\geq f(B^{m-1}, m - 1) - p(x + 1 - B^{m-1}) \\
= T_F f(x + 1, m).
\]

**\( T_R \):**

\[
T_R f(x, m) = f(x, m + 1) \\
\geq f(x + 1, m + 1) \\
= T_R f(x + 1, m).
\]

Recall that \( \lambda + B^M \mu + \sigma + \tau = 1 \) and \( \beta < 1 \). Thus, by the above inequalities, we have that \( T f(x, m) \geq T f(x + 1, m) \).

The following technical lemma is required for the proof of the next Theorem.

**Lemma 5.4** For all \( 0 \leq x + 1 \leq B^m, 0 \leq m \leq M \) and \( n \), we have that

\[
V_n(x + 1, m) \geq V_n(x, m) - p. \tag{5.6}
\]
Proof. Lemma 5.4 follows by induction on $n$. It is easily seen that $V_0$ (i.e., $-C$) satisfies the above inequality for all $0 \leq x \leq m - 1$. We show that, if $f(x,m)$ satisfies inequality (5.6), then so does $T_* f(x,m)$ for all event operators $T_*$. The terms concerning the operators $T_A$ and $T_{MD}$ are similar to those in Lemma 5.2. Consider, then, the operators $T_F$ and $T_R$.

$T_F$: Considering $x + 1 \leq B^m - 1$, it follows that

$$T_F f(x + 1, m) = f(x + 1, m - 1) \geq f(x, m - 1) - p \geq T_F f(x, m) - p.$$ 

Now, for $x \geq B^{m - 1}$, we have:

$$T_F f(x + 1, m) = f(B^{m - 1}, m - 1) - p(x + 1 - B^{m - 1}) = T_F f(x, m) - p.$$

$T_R$:

$$T_R f(x + 1, m) = f(x + 1, m + 1) \geq f(x, m + 1) - p = T_R f(x, m) - p.$$

Recall that

$$T f(x,m) = -C(x,m) + \beta [\lambda T_A f(x,m) + \mu T_{MD} f(x,m) + \sigma T_F f(x,m) + \tau T_R f(x,m)],$$

$$\lambda + B^M \mu + \sigma + \tau = 1$$

and $\beta < 1$. Thus, by the above inequalities, we have that $T f(x + 1, m) \geq T f(x, m) - p$. 

The following theorem shows that the optimal admission policy uses the idea of trunk reservation, i.e., for each channel state there is a threshold above which no connections are admitted. The thresholds assure that some bandwidth is kept unutilized to reduce the probability of connection termination upon a decrease in the wireless channel bandwidth.

**Theorem 5.2**

1. For all $0 \leq x + 2 \leq B^m, 0 \leq m \leq M$ and $n$, we have that

$$2V_n(x + 1, m) \geq V_n(x, m) + V_n(x + 2, m),$$

i.e., $V_n$ is concave with $x$. 

126
2. The optimal policy is of threshold form, i.e., for each channel state there is a critical level above which no connections are admitted.

Proof. It is easily seen that \( V_0 \) (i.e. \(-C\)) satisfies (5.7). We show that, if \( f(x,m) \) satisfies (5.7), then so does \( T_* f(x,m) \) for all event operators \( T_* \in \{ T_A, T_{MD}, T_F, T_R \} \). As the inequality is closed under linear combinations, the lemma follows by induction on \( n \).

The concavity of \( T_A \) with respect to \( x \) follows from the same arguments as in the proof of Theorem 5.1. Consider, then, \( T_F \). The result follows easily for \( x+1 < B^m-1 \) and for \( x+1 > B^m-1 \). Considering \( x+1 = B^m-1 \), we have that:

\[
2T_F f(x+1, m) = 2f(x+1, m-1) \\
\geq f(x, m-1) - p + f(x+1, m-1) \\
= T_F f(x, m) + T_F f(x+2, m),
\]

where the inequality is derived from Lemma 5.4.

The terms concerning \( T_{MD} \) and \( T_R \) are straightforward and the first part of the theorem follows.

By the first part of the theorem, we have that, for any \( m \leq M \),

\[
r + V_n(x+1, m) - V_n(x, m) \geq r + V_n(x+2, m) - V_n(x+1, m).
\]

Thus, if \( r + V_n(x+2, m) - V_n(x+1, m) \geq 0 \), i.e., admission is optimal in state \((x+1, m)\), then also \( r + V_n(x+1, m) - V_n(x, m) \geq 0 \), i.e., admission is optimal in state \((x, m)\). Similarly, we can show that, if rejection is optimal in \((x, m)\), then so it is in state \((x+1, m)\).

5.4 Numerical Examples

In this section we numerically study the structure of the optimal policy and validate the results obtained in Sections 5.2 and 5.3. That is, our numerical study validates the optimality of the trunk reservation admission policy. The expected discounted reward value was evaluated through the value iteration procedure.

Example 5.1 We study the model of Section 5.2 with (i) \( M = 20 \) resources, (ii) the following rate parameters \( \lambda = 1000, \mu = 100, \sigma = 1, \tau = 10 \), and (iii) the costs (rewards) \( r = 1, p = 100 \). The computation ends when the variation of the value function \( V_n \) between two iterations is less than \( \epsilon = 0.0000001 \). The discount factor is \( \beta = 0.99 \). Figure 5.3 represents the obtained optimal policy. A circle means acceptance of connections. The \( x \) axis represents the connections and the \( y \) axis represents the resources. Indeed, in accordance with Theorem 5.1, the admission policy depicted in the figure is of the threshold type. Also, note that arriving connections might be rejected even when there are available resources, e.g., at state \((19, 20)\).
Example 5.2 In this example we consider the model of Section 5.3. We assume that the supported rates are according to IEEE 802.11g (i.e., 1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48 and 54 Mbit/s) and that each connection consumes a bandwidth of 1 Mbit/s. Furthermore we assume the rate parameters $\lambda = 100$, $\mu = 10$, $\sigma = 100$, $\tau = 100$, and the costs (rewards) $r = 1$, $p = 2$. The discount factor is $\beta = 0.99$. Figure 5.4 represents the optimal policy obtained. The $x$ axis represents the connections and the $y$ axis represents the 802.11 rates. As in Example 5.1, and in accordance with Theorem 5.2, the admission policy obtained numerically is of the threshold type. Also, here too, arriving connections might be rejected even when there are available resources, e.g., at state (40, 54), i.e., consumed bandwidth of 40 Mbit/s and available bandwidth of 54 Mbit/s.
5.5 Conclusion

We considered a discrete model for optimal call admission control in fault tolerant networks with varying number of resources. This model is proposed for both optical as well as wireless networks. In this context, termination of ongoing calls due to failures is more “expensive” than rejection of arriving calls. Accordingly, the network controller may reserve some of the network resources for restoration. We employed event-based dynamic programming to solve the admission control problem, and we established the optimality of the trunk reservation policy. We further validated the results numerically.

Throughout this paper, we have assumed a single class of connections. An important direction for future research is to consider connections of different classes with different bandwidth (alternatively, different number of resources) requirements.
Chapter 6

Conclusion

This dissertation has studied QoS provision, call admission control and routing under various settings. Specifically, it has investigated the establishment of end-to-end performance bounds and the corresponding routing problems as a whole, under both the deterministic as well the stochastic settings. Furthermore, it has studied the call admission control problem in fault tolerant networks.

In Chapter 2, we studied networks that employ the GPS scheduling discipline. While previous studies (e.g., [8, 9, 10]) exclusively dealt with (deterministically) burstiness constrained (BC) traffic, deterministic guarantees and fixed-rate links, we (mainly) investigated the problem within the realm of stochastically bounded burstiness and stochastic guarantees; moreover, we also investigated the important case of variable-rate links. First, in order to establish appropriate routing schemes, we needed to have at hand end-to-end delay bounds for networks of packetized servers and links with non-negligible propagation delays. Since previous work on stochastic (EBB, SBB) settings have been carried on more limited models (at times, on a single, isolated server), we had to make the required extensions. With these bounds at hand, we established QoS routing schemes that are designed to operate in conjunction with the GPS scheduling discipline. The new bounds have a much more complex structure than the deterministic bound of the “basic” BC setting. Moreover, the way they should be employed within a corresponding routing scheme is not as straightforward as in the basic setting. Yet, the complexity of the resulting QoS routing scheme is typically not higher than in the basic setting. After having established the routing schemes, we tested them by way of simulation.

In Chapters 3 and 4, we turned to consider the EDF scheduling discipline. First, in Chapter 3, we considered BC traffic and deterministic guarantees. Then, in Chapter 4, we considered EBF traffic and stochastic guarantees.

Under the deterministic setting, we studied both the basic problem of identifying feasible paths as well as the more complex problem of optimizing the route choice in terms of balancing the loads and accommodating multiple connections. First,
we broadened the space of feasible solutions by allowing to reshape the traffic with different parameters at each hop. Consequently, we introduced both optimal as well as computationally efficient $\epsilon$-optimal routing schemes that considered the joint problem of identifying a feasible path and optimizing the reshaping parameters along the path. Next, we established and validated two routing schemes that aim at balancing the load and optimizing the deadline assignment along the chosen path. Simulation results demonstrated the advantages of our schemes.

Under the stochastic setting, we analyzed both the single node case, which consists of an EDF scheduler and an EBB traffic shaper, as well the multiple node case. Consequently, we have provided, for the first time, end-to-end bounds for EBB traffic and systems of EDF schedulers. Moreover, we have shown that any upper bound on the end-to-end delay tail distribution that can be guaranteed by the GPS discipline, can also be guaranteed by the EDF discipline.

Finally, in Chapter 5, we considered the provision of fault tolerance, which emerges as an important QoS requirement. Here, we focused on call admission control policies. Specifically, we proposed a discrete call admission control model for both optical as well as wireless networks. In this context, termination of ongoing calls due to failures is more “expensive” than rejection of arriving calls. Accordingly, the network controller may reserve some of the network resources for restoration. We employed event-based dynamic programming to solve the admission control problem, and established the optimality of the trunk reservation policy. We further validated the results numerically.

In addition to the specific summaries of future research directions given in the concluding sections of each chapter, we now consider more general directions for extending our work.

6.1 Non-worst-case analysis

The current research, as most work on scheduling, has focused on worst-case delay bounds. However, this approach necessarily leads to conservative bounds, since they must allow for the possibility that bursts from all the sources incident to a server arrive simultaneously. In another approach, known as statistical multiplexing, all the sources are assumed to be independent, which implies that, for a large number of sessions, such simultaneous bursts are extremely unlikely. This enables the establishment of much tighter delay bounds that hold with high probability. Several works analyzed statistical multiplexing in the context of both the GPS [35, 34] and the EDF [44, 45] scheduling disciplines. A key concept in such an analysis is the effective bandwidth of a session, i.e., the minimum rate that a source needs to meet its QoS requirements. Here, a possible direction for future research is to
investigate previously established statistical end-to-end delay bounds, attempt to simplify the mathematical expressions so as to derive routing metrics, and establish the corresponding routing schemes.

6.2 Hierarchical networks and topology aggregation

A challenge of our study is to further enhance the scalability of its proposed schemes. A powerful tool for providing scalability in large scale networks is that of topology aggregation. We proceed to discuss some specific ideas in that respect.

In hierarchical routing, nodes and links at each hierarchical level may be recursively aggregated into higher levels. At the lowest level of the hierarchy, each node represents a switching system. At a higher level, each node represents a collection of one or more nodes at a lower level. Similarly, each link at the lowest level represents a physical link, whereas at a higher level, each link represents a connectivity formed by one or more lower level links in series and/or in parallel. Hierarchical routing is employed, for example, by the PNNI scheme [61].

End-to-end delay bounds established for such hierarchical networks should account for the actual number of hops, i.e., the number of physical links from the lowest level of the hierarchy, rather than the number of links along the path. Obviously, the number of hops associated with each link should be known to the router, and be considered when a routing decision is made. In [62] we considered a general model, suitable for hierarchical networks, where each link is composed of one or several hops. We showed that the routing algorithms proposed in Chapter 2, with slight modifications, can be efficiently applied in hierarchical networks. However, since the actual number of hops, which is required for the routing schemes, is not always known to the source, an estimation of this value is desirable. A reasonable assumption is that the link delay and bandwidth are highly correlated with the number of actual hops. Thus, the number of hops can be estimated from these values. Accordingly, an efficient scheme that estimates the number of hops under realistic assumptions is called for.

Consider now the establishment of topology aggregation schemes. A possible direction for future work is to consider the employment of rate-quantization for joint aggregation of link delay and bandwidth. Most studies on topology aggregation have considered the aggregation of each metric separately. However, since the proposed QoS routing algorithms in Chapter 2 seek the shortest path with respect to the delay for several rate values, advertising the delay of each aggregated link for these values is sufficient. Furthermore, when rate-quantized routing schemes are employed, advertising the delay for each quantization level is sufficient. One can see that
uniform quantization may lead to excessive topology advertisement, thus improved quantization schemes are called for.

As topology aggregation is a process of summarizing and compressing (topological) information, a possible research direction is to employ Information Theory in order to establish efficient compression schemes. In [63], the problem of compressing a data structure (e.g. tree, undirected and directed graphs) in an efficient way while keeping a similar structure in the compressed form was investigated. There, the idea of building an LZW tree by way of LZW compression is used to compress a binary tree generated by a stationary ergodic source. LZW compression is a popular version of Lempel-Ziv (LZ) algorithm [64] and has been implemented as the “compress” command on UNIX systems. In view of the encouraging results of [63], future research should investigate the use of similar compression algorithms as the basis for a topology aggregation process. Still, there are several obstacles to be resolved. First, the proposed compression algorithm in [63] considers only the structure of the graph/tree, but does not consider any metrics associated with each link (e.g., delay, bandwidth). Considering these metrics while building the LZW dictionary may dramatically reduce the efficiency of the compression scheme. Next, while the proposed tree compression scheme achieves optimal compression, the graph compression scheme may not always yield a compressed form. Finally, with the compressed graph together with the dictionary at hand, reconstructing the actual network from the compressed form, for routing purposes, may be a complex task.

Alternatively, we may characterize a graph as an “image”: establishing a proper transformation between graphs and images may facilitate the use of image processing and compression schemes as graph compression algorithms.
Bibliography


