QoS Provision and Routing with Deterministic and Stochastic Guarantees

Ph.D. Research Proposal

Erez Biton

supervised by
Prof. Ariel Orda

April 1, 2001
Abstract

We consider QoS provision and routing schemes for connections with end-to-end delay requirements in networks which employ rate-based schedulers. First, we consider 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 an end-to-end delay bound on the tail distribution for packetized traffic and links with non-negligible propagation delays. With this bound at hand, 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 too, 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. Here, we consider two settings: a deterministic setting, of links with fluctuation constraints and BC traffic, and a stochastic setting, of links with exponentially bounded fluctuation and EBB traffic. In both 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.


1 Introduction

Emerging broadband high speed networks are expected to support real time and multimedia applications with various quality of service (QoS) requirements. QoS architecture should provide these 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.

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, as well as 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 then, to find a path which complies with end-to-end constraints derived from the QoS users needs. Obviously, QoS routing is an essential part in QoS provision. Lack of good QoS routing algorithms may cause problems such as high call blocking probability, low network utilization and long connection setup delays.

Scheduling disciplines for guaranteed performance as well as worst-case end-to-end performance bounds have been established and explored by a large number of works, both under the deterministic and stochastic settings (e.g., [1, 2, 3, 4, 5, 6] and references therein). Recently, the corresponding routing problems have been addressed as well (e.g., [7, 8, 9, 10]). However, a comprehensive study which considers both problems, i.e., the establishment of end-to-end performance bounds and the corresponding routing problem, have not been reported. Since these two problems are closely related, in fact the first is a prerequisite for the second, we believe they should be investigated concurrently. For example, the generalized processor sharing scheduling discipline has been studied in [6] under the stochastic settings and worst-case bounds on the end-to-end delay tail distribution have been established. However, these bounds cannot be utilized by routing algorithms since they do not account for packetized traffic, nor for propagation delays. Furthermore, corresponding routing algorithms (to our best knowledge) have not been established.

Thus, the establishment (or extension) of bounds on the end-to-end backlog and delay for various settings (e.g., traffic models, scheduling disciplines, hierarchical networks, etc.), as well as the development of corresponding efficient QoS routing schemes, are the goals of this research.

The current study focuses on the "rate based" class [1, 4] and in particular on the Generalized Processor Sharing - GPS [2, 3] (also known as Weighted Fair Queuing) scheduling discipline. 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.

In [1, 3, 4], networks with rate-based schedulers were studied under a deterministic setting, and worst-case bounds on the end-to-end delay were derived. The corresponding routing problem, of identifying the best path with respect to the QoS delay requirements, has been the subject of several studies, e.g. [7, 8, 9, 11, 12]. In particular, it was shown in [7, 9] that for a given connection with 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. In [8], 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 the paper.

Under a deterministic setting, the input traffic bursts are assumed to be of bounded length. This is not the case in most commonly used input processes. 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. 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 former routing problem, which considered 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 worst-case 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.

Finally, we consider networks with variable-rate links. In [15], 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 and the stochastic settings.

The rest of the paper is structured as follows.

2 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 short survey of proposed scheduling disciplines, focusing on those that are most relevant to the current research. Next, we deal with the establishment of end-to-end delay bounds. Then, we consider the corresponding routing problems. Finally, we briefly discuss two important concepts that are relevant to the current and future research, namely effective bandwidth and hierarchical networks.

2.1 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 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 [16] 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 [17] 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 Bound Burstiness, 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:

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

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:

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

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 \((\int_0^\infty dt)^n f(t)\) 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 [18]: 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 [19]. 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 $\epsilon_e(d)$ if, for all non-negative $d$, it holds that

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

Finally, we mention the $(X_{min}, X_{ave}, I, S_{max})$ traffic model, introduced in [20]: here, $X_{min}$ is the minimum inter-arrival time between any two packets in the stream, $X_{ave}$ is the average packet inter-arrival time over an averaging interval, $I$ is the length of the interval, and $S_{max}$ is the maximum packet size.

### 2.2 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, 21], and in particular the Generalized Processor Sharing (GPS) [2, 3] (also known as Fluid Fair Queueing) scheduling discipline. 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 $\phi_1, \phi_2, \ldots, \phi_m$. 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 which

$$\frac{S_i(\tau, t)}{S_j(\tau, t)} \geq \frac{\phi_i}{\phi_j}, \quad j = 1, 2, \ldots, m$$

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

$$g_i \triangleq \frac{\phi_i}{\sum_j \phi_j} R.$$ 

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]:

$$S_i(0, \tau) - \hat{S}_i(0, \tau) \leq L_{max}.$$ 

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) [22]. In [23], 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 [15].

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. Several non-work-conserving disciplines have been proposed, among them are: Earliest-Deadline-First (EDF) [20], Hierarchical Round Robin (HRR), and Static Priority (SP) [24].

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].

An SP scheduler distinguishes among \( P \) priority levels and maintains one FIFO queue for each priority. Each connection is assigned a priority \( p \) with \( 1 \leq p \leq P \), and packets arriving on a connection are inserted into the FIFO queue corresponding to the connection’s priority. At the beginning of a busy period, or after completing the transmission of a packet, the SP scheduler always selects the first packet in the nonempty FIFO queue with the highest priority for transmission.

### 2.3 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, 16, 25]), or whether they integrate individual servers in the network into larger super-servers (the service-curve approach [2, 3, 18]).

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 GPS servers:

\[
D_i \leq \frac{\sigma_i + n(p) L_i}{r_i(p)} + \sum_{l \in P} \left( \frac{L_{\text{max}}}{R_i} + d_l \right),
\]

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 \( p \), \( R_i \) 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 exist 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 \( \frac{\phi_i}{\phi_j} < \frac{\rho_i}{\rho_j} \). The special case for which \( \phi_i = \rho_i \) for every session \( i \), is called Rate Proportional Processor Sharing (RPPS).

Closed-Form deterministic end-to-end delay bounds for the broader class of CRST GPS networks were established in [26]. 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 \( D > 0 \),

\[
\Pr \left\{ D_i^{(p)}(t) \geq D \right\} \leq \Lambda_i^{(p)} e^{-\alpha_i r_i(p) \cdot D},
\]

where

\[
\Lambda_i^{(p)} = \frac{\Lambda_i \cdot e^{\alpha \cdot \rho_i \cdot \xi}}{1 - e^{-\alpha \cdot (r_i(p) - \rho_i) \cdot \xi}}, \quad 0 < \xi < \frac{\ln (\Lambda_i + 1)}{\alpha \cdot (r_i(p) - \rho_i)},
\]

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 [19]. There, stochastic bounds on delay and backlog of a network element in isolation were derived for a \( \text{b-smooth} \) with earliness profile \( \epsilon \) input traffic. A network analysis (the multiple node case) was not provided, neither were bounds for the simpler \( \text{EBB} \) 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 \textit{rate-controlled} service disciplines [21]. 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 [21]. 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 \textit{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 [27].

2.4 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 [7, 8, 9, 11, 12].

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 NP-hard (i.e.,), hence, in general, optimal solutions of polynomial complexity ******. 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 [28]. In [11] 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. [7, 8, 9, 12]. In particular,
it was shown in [7] 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 [9]). In [8], 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 must have available schedulability conditions that detect violations of delay guarantees in a network switch. Exact schedulability conditions for EDF and SP disciplines have been presented in [29]. Call admission control algorithms for the RC-EDF service discipline have been studied in [30]. Although the EDF schedulability condition in [29] 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 [30], 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 [10, 20, 31].

In [20], 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 will be 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 [10], 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) [31]. 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 [10] it is proposed to divide the "extra" delay along the route, either evenly or in proportion to each link’s MWRT.

### 2.5 Effective Bandwidth

The complexity of QoS provision in high speed networks with heterogeneous applications, as opposed to traditional telephone networks, lies in the multiplexing of multiple types of packetized traffic streams via switches and communication links. To ease the task of managing such networks, it is desirable to characterize, as a function of the required QoS, the effective bandwidth requirement of both individual connections and the aggregate bandwidth usage of connections multiplexed on a given link [32]. The effective bandwidth (or equivalent capacity) of a set of connections multiplexed on a link is defined as the amount of bandwidth required to achieve a desired QoS, e.g., buffer overflow probability, given the offered aggregated bit rate generated by the connections. Specifically, suppose a stringent of the form

$$\Pr \{ W > B \} \leq e^{-\delta B}$$

is to be satisfied by the workload $W$ of a shared buffer size $B$. For an asymptotically large $B$, one can show that

$$\sum_{j \in J} \alpha_j (\delta) \leq r \iff \lim_{B \to \infty} \frac{1}{B} \log \Pr \{ W > B \} \leq -\delta,$$

where $r$ is the link bandwidth and $\alpha_j (\delta)$ is the effective bandwidth of stream $j$. With the effective bandwidth at hand, call admission can be easily carried out by checking if the available capacity is larger than the effective...
bandwidth of the new stream.

2.6 Hierarchical Networks

As network grow in size, it becomes impossible to maintain exact information on the entire network. One possible solution, which was accepted by the ATM Forum PNNI standard [33], is to introduce a hierarchical process that progressively “aggregates” state information as networks get more and more remote. Accordingly, nodes and links at a hierarchical level may be recursively aggregate into higher levels. At the lowest level of the hierarchy, each node represent a switching system. At each other level, each node represents a collection of one or more nodes at a lower level. The process of summarizing and compressing topology information at each hierarchical level, in order to determine the topology information to be advertised at the collection of one or more nodes at a lower level. The process of summarizing and compressing topology information at each hierarchical level, in order to determine the topology information to be advertised at the level above, is referred to as topology aggregation. In [34], several methods for topology aggregation have been described, among them the full-mesh, symmetric-node, and star approaches. In [35], a spanning tree approach for link state aggregation has been proposed, which considers link states that are composed of link delay and bandwidth. The hierarchical routing problem has been the subject of several studies, e.g. [7, 8]. In [7], routing in networks with inaccurate information which may result from topology aggregation was investigated. In [8], a near-optimal algorithm for the restricted shortest path problem in hierarchical networks has been established.

3 Current Results

3.1 Model Formulation

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, and let $|V| = N$ and $|E| = M$.

Following [7, 9, 8], 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.

Denote the the service rate of a link $l$ by $R_l^i$. Let $I(l)$ denote the set of sessions present at link $l$. For $i \in I(l)$, the guaranteed service rate for session $i$ at a link $l$ is $r_l^i$. Let the available rate values in the network be $R_1, R_2, \ldots, R_K$, where $R_1 \leq R_2 \leq \ldots \leq R_K$ and $K \leq M$.

Each link $l \in E$ is characterized by a constant delay $d_l$, related mainly to the propagation delay.

A session $i$ is routed through a path $p(i)$. The $n$-th hop ($n$-th server) in the path $p(i)$ is denoted by $p_n(i)$. Let $n(p(i))$ be the total number of hops along $p(i)$. We denote by $H$ the maximal possible number of hops along a path.

Let $r_i(p)$ be the guaranteed service rate for session $i$ along $p$, i.e. $r_i(p) = \min_{\nu \in p(i)} r_{i\nu}$, where $r_{i\nu}$ is the guaranteed service rate at server $\nu \in p(i)$. For every session $i \in I(\nu)$, denote the arrival process into a server $\nu$ by $A_{i\nu}$ and the departure process by $S_{i\nu}$. Let $A_i$ be the arrival process of session $i$ into the first hop on its path, $p_1(i)$, and $A_i(n)$ be the arrival process at the $n$-th hop, $p_n(i)$. Thus, when $\nu = p_n(i)$ for a particular session $i$, the processes $A_{i\nu}$ and $A_i(n)$ are identical. The session $i$ departure process at the $n$-th hop is denoted by $S_i(n)$, and $S_i_{out} = S_{i(n(p(i))}$ describes the network departure process of session $i$. The backlog of session $i$ at node $p_n(i)$ at time $t$ is denoted by $Q_i^{(n)}(t)$, and the total backlog over the path $p$ at time $t$ by $Q_i^{(p)}(t)$. Finally, the end-to-end delay of session $i$ at time $t$ is denoted by $D_i^{(p)}(t)$.

3.2 QoS Provision and Routing with Deterministic Guarantees

In this section we consider QoS provision and routing under a deterministic setting. First we extend the results presented in [8] and provide some foundations which facilitate the presentation in the next sections. Next, we consider networks with variable-rate links, here, we derive bounds on the end-to-end delay, and propose corresponding routing algorithms.
3.2.1 Near-optimal QoS routing algorithms

In [8], optimal solutions have been 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. 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 \( r(p) \geq \rho \). Denote by \( r_{\text{min}}(p) \) the minimal value of \( r \) for which the path is feasible, i.e., \( D(p,r_{\text{min}}(p)) = D \) and \( \rho \leq r_{\text{min}}(p) \leq r(p) \). 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 \( \hat{R} = \frac{R^K}{R} \). As in [8], 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 \( R^1 (1 + \epsilon)^j \ldots R^K (1 + \epsilon)^{j+1} \). We say that a link \( l \) is in rate-class \( j \) if the value of \( r^l \) is in that class rate.

3.2.1.1 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.

3.2.1.1.1 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 \( O \left( M \cdot H \cdot K \right) \), was proposed in [8]. In [36] we formulate a rate-quantized solution that finds a near-optimal path with complexity of \( O \left( M \cdot H \cdot \min \left\{ \frac{1}{\epsilon} \cdot \log \hat{R}, K \right\} \right) \), where \( \hat{R} = \frac{R^K}{R} \). Due to space constraints, we present here just a sketch of the algorithm.

3.2.1.1.2 Minimum Rate - Rate Quantized (MR-RQ) - sketch: For each \( j \), \( 0 \leq j \leq \lfloor \log_{1+\epsilon} \hat{R} \rfloor \), perform the following: delete all links whose rate-class is less than \( j \); on the remaining graph, execute a Bellman-Ford shortest path algorithm, with respect to the metric \( \{ d_l \} \), in order to identify a shortest path for each possible number of hops \( n \), for all \( 1 \leq n \leq H \). This way, \( H \) paths are identified at each iteration, and \( O \left( \frac{1}{\epsilon} \cdot \log \hat{R} \cdot H \right) \) paths overall. The solution is found by computing the respective minimal rate of each such path, and choosing the best among them.

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 \(\frac{1}{e} \cdot \log \hat{R} > K\).

### 3.2.1.2 Load Balancing Criterion

A better measure for balancing loads over the network may be the relative (rather than absolute) rate consumption \([8]\). The problem may be stated as finding a feasible path \(p\) that minimizes the value of \(\max_{l \in p} \frac{r_{\min}(p)}{r_l}\). The algorithm Minimum Relative Rate (MRR), introduced in \([8]\), solves this problem with complexity of \(O(M \cdot H \cdot K)\). In \([36]\) we employ the rate-quantization method to achieve a near-optimal solution with complexity of \(O(M \cdot H \cdot \min \left\{ \frac{1}{e} \cdot \log \hat{R}, K \right\})\).

### 3.2.2 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 \([15]\). Some typical examples are shared-media links governed by MAC protocols, flow controlled links, and wireless 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 \([7]\). Motivated by \([15]\), 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 be Fluctuation Constrained (FC) with parameters \((\Delta, r)\), if for all \(s, \tau (\tau > s > 0)\):

\[
S(s, \tau) = \int_s^\tau S(t) \, dt \geq \left[ r(\tau - s) - \Delta \right]^+ \tag{3}
\]

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

We apply the results of \([15]\) 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 1.** Let \(I = \{1, 2, \ldots, m\}\) be the set of sessions entering an RPPS GPS server with a variable service rate. Let the service rate be FC with parameters \((\Delta, R_I)\), and let the input traffic be BC. For any session \(i \in I\), the backlog and delay are bounded as follows:

\[
Q_i(t) \leq \sigma_i + \frac{\rho_i}{r_i} \Delta_i, \tag{4}
\]

\[
D_i(t) \leq \sigma_i + \frac{\Delta_i}{r_i}, \tag{5}
\]

where

\[
\Delta_i \triangleq \frac{\rho_i}{\sum_{j=1}^n \rho_j} \cdot \Delta, \quad r_i \triangleq \frac{\rho_i}{\sum_{j=1}^n \rho_j} \cdot R_I.
\]

\[\Box\]

Next, consider a network with variable-rate RPPS GPS servers. Proposition 2 implies that the end-to-end delay in a network with variable-rate RPPS GPS servers equals to the delay imposed by a single FC server with parameters \(\left( \sum_{n=1}^{n(p(i))} \Delta_i^{(n)}, r_i(p) \right)\). This result is similar to the one obtained with constant rate links.

**Proposition 2.** For every session \(i\) with BC traffic in an RPPS GPS network with FC service rate, we have:

\[
Q_i^{p}(t) \leq \sigma_i + \frac{\rho_i}{r_i(p)} \sum_{n=1}^{n(p(i))} \Delta_i^{(n)}, \tag{6}
\]
When packet sizes \((L)\) and propagation delays \((d_i)\) are not negligible, the following proposition holds.

**Proposition 3.** For each session \(i:\)

\[
D(p, r) \leq \sigma + \frac{\sum_{l \in p} n_l \cdot L + \sum_{l \in p} \Delta_l}{r} + \sum_{l \in p} d_l.
\]

3.2.2.1 Routing algorithms

A key observation in the design of corresponding QoS routing schemes is that the rate fluctuation effect on the end-to-end delay bound can be combined with the non-cut-through effect. Specifically, let \(x_l = L + \Delta_l\); then,

\[
D(p, r) \leq \sigma + \frac{\sum_{l \in p} x_l}{r} + \sum_{l \in p} d_l.
\]

It is clear that the routing schemes introduced in [8] and in Section 3.2 can be applied to variable-rate links as well.

To illustrate it, consider, for example, the basic problem of identifying a feasible path that minimizes the end-to-end delay bound. Such a path can be identified through \(K\) executions of Dijkstra’s shortest path algorithm with respect to the metric \(\{x_l + d_l\}\), for all possible maximal rates, i.e., \(r = R^1, R^2, \ldots, R^K\). Hence, the algorithm’s complexity is \(O((N \cdot \log N + M) \cdot K)\). Obviously, rate quantization can be applied here too to reduce the complexity.

3.3 QoS Provision and Routing with Stochastic Guarantees

In this section we turn to consider stochastic settings, starting with EBB processes: first we establish bounds on the end-to-end delay distribution for a PGPS network with non-negligible propagation delays; then, we propose corresponding routing schemes. 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. Finally, we consider networks with variable-rate links under a stochastic setting.

3.3.1 EBB Traffic

In Section 3.2 we considered routing schemes under a deterministic setting. However, many applications, in particular multimedia applications, can tolerate a certain amount of fluctuations beyond the guaranteed delay. For such applications, a better approach is to identify a route with stochastic end-to-end guarantees. Under a stochastic setting, the traffic bursts are not required to be firmly regulated; rather, their probability distributions are bounded. Therefore, stochastic guarantees can be provided for a larger class of input processes. Another important advantage of the stochastic setting is the ability to provide tighter bounds, thus achieving better network utilization.

In this section, we adopt the model introduced in [13] of Exponentially Bounded Burstiness (EBB) processes, and consider networks of RPPS PGPS servers. 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 the rate quantization method to the stochastic setting.
3.3.1.1 Upper Bound on the End-to-End Delay Tail Distribution  In [36] we extend the results of [6] to Packetized traffic and establish upper bounds on the end-to-end delay tail distribution for a PGPS network.

When packet size is not negligible, there are two effects to be considered. First, packets are served non-preemptively, i.e., once the server has begun serving a packet, it continues to do so until completion. Second, the service is non-cut-through, i.e., no packet is eligible for service until its last bit has arrived. Accordingly, when these effects are considered, the following upper bound on the end-to-end delay tail distribution holds.

**Proposition 4.** For every session with EBB traffic in an RPPS PGPS network, for any $D > 0$, the end-to-end delay tail distribution is upper bounded as follows:

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

(9)

Denote this end-to-end delay tail distribution upper bound by $\mathbf{Fr}(D, p)$.

3.3.1.2 Routing Algorithms  With the above upper bound (9) 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 cannot be immediately applied, hence new schemes are called for. Under the stochastic setting, the end-to-end delay, over all possible paths, cannot be guaranteed not to exceed any value. Thus, even the basic problem of identifying a feasible path should be revised. Moreover, a new definition for path feasibility is required.

We assume that each application is associated with a certain probability $q$, which reflects its “sensitivity” to end-to-end delay fluctuations beyond the required delay. Intuitively, a path is said to be feasible if the end-to-end delay fluctuations beyond the required value conforms with the applications “sensitivity”.

**Definition 1.** Define the effective end-to-end delay $D_{\text{eff}}(p, q)$ of a path $p$, as the maximal end-to-end delay for which the tail distribution is guaranteed to be less or equal to $q$, i.e., $\Pr \{ D(p) \geq D_{\text{eff}}(p, q) \} \leq q$.

**Definition 2.** A path $p$ is $q$-feasible if the end-to-end delay tail distribution beyond the required delay $D$, is lower or equal to $q$, i.e., $\Pr \{ D(p) \geq D \} \leq q$. Alternatively, a path $p$ is $q$-feasible if the required delay $D$ is higher or equal to the effective delay $D_{\text{eff}}(p, q)$ of the path $p$, i.e., $D \geq D_{\text{eff}}(p, q)$.

Note that a similar idea is used in the definition of effective bandwidths. There, the effective bandwidth is defined as the amount of bandwidth required to achieve a desired QoS, e.g., buffer overflow probability.

3.3.1.2.1 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:

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

The following algorithm identifies such a path.
Algorithm Minimum Delay tail Distribution (MDD)

1. For \( k \leftarrow 1 \) 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) 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, \( p(k) \) \( 1 \leq k \leq K \).

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

Proposition 5. Algorithm MDD correctly identifies the \( q \)-feasibility of the connection. Whenever the connection is \( q \)-feasible, the path \( p(\tilde{k}) \), identified by the algorithm, achieves the minimal delay tail distribution upper bound \( \Pr(D, p(\tilde{k})) \) among all \( q \)-feasible solutions. The algorithm’s complexity is \( O((N \log N + M) \cdot K) \).

3.3.1.2.2 Minimum Cost Path  
Next, we consider the problem of finding a feasible path that optimizes some general cost function. Following [8], 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 \( C(r, p) = C(r, n(p), D(p, q)) \) and it is nondecreasing in each of its three arguments. In other words, the cost is a nondecreasing function of (i) the number of hops \( n(p) \), (ii) the consumed rate \( r \), and (iii) the effective end-to-end delay \( D_{eff}(p, q) \).

The following algorithm identifies an optimal path, i.e., with minimal cost.

Algorithm Minimum Cost (MC)

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

Proposition 6. 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 \( O(M \cdot H \cdot K) \).
3.3.1.2.3 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:

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

Such a path can be identified through \( K \) executions of Dijkstra’s shortest path algorithm with respect to the metric \( \{ \frac{L}{r} + d_l \} \). Thus, the routing problem can be solved with complexity of \( O \left( (N \log N + M) K \right) \). While being polynomial, such complexity could still be prohibitive [8].

Hence, we propose a rate quantized scheme, which identifies a near-optimal solution with lower complexity. Applying the standard (uniform) rate quantization method, as illustrated under a deterministic setting in Section 3.2, leads to an \( \epsilon \)-optimal solution with complexity of

\[
O \left( (N \log N + M) \cdot \min \left\{ \frac{2R^1 - \rho}{(R^1 - \rho) \epsilon} \cdot \log \hat{R}, K \right\} \right).
\]

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 \( R(j) \leq \alpha r^l < R(j + 1) \), 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.

Then, the Rate-Quantized Quickest path algorithm is formulated as follows.

Algorithm Quickest Path - Rate Quantized (QP-RQ)

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 \( \left\{ \frac{L}{\pi(j)} + d_l \right\} \) through Dijkstra’s shortest path algorithm.
   (d) Compute \( D_{\text{eff}}(p(j), q) \).
   (e) If \( R(j) < \rho (\Lambda + 1) \) then \( a_{j+1} \leftarrow \frac{\rho (1+\epsilon)^{j+1}}{\rho (1+\epsilon)^{j+1}} \left( R^1 - \rho \right) \), else \( a_{j+1} \leftarrow \frac{(1+\epsilon)^{j+1} - 1}{(1+\epsilon)^{j+1} - 1} \Lambda \).
   (f) \( R(j + 1) \leftarrow a_{j+1} \cdot R(j) \).
3. Among all paths \( p(j) \), choose a path \( \tilde{p}(\tilde{j}) \) with the minimum effective delay \( D_{\text{eff}}(p(\tilde{j}), q) \).

Proposition 7.

1. 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 \( D_{\text{eff}}(p(\tilde{j}), q) \leq (1 + \epsilon) \cdot D_{\text{eff}}(p^*, q) \).
2. The algorithm’s complexity is

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

3.3.2 SBB Traffic

We consider a more general source traffic model, of 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 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.

3.3.2.1 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 8.** 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 $\xi > 0$,

$$\Pr\{Q_i(t) \geq Q\} \leq f\left((Q - \rho_i \cdot \xi_i)^+\right) + \frac{1}{(r_i - \rho_i) \cdot \xi_i} \int_{Q-\rho_i \cdot \xi_i}^{\infty} f(u) \, du,$$

and

$$\Pr\{D_i(t) \geq D\} \leq f\left((r_i \cdot D - \rho_i \cdot \xi_i)^+\right) + \frac{1}{(r_i - \rho_i) \cdot \xi_i} \int_{r_i \cdot D - \rho_i \cdot \xi_i}^{\infty} f(u) \, du,$$

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

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

**Proposition 9.** For every session $i$ with SBB traffic in an RPPS PGPS network, for any $\xi > 0$, $Q > 0$ and $D > 0$:

$$\Pr\{Q_i^{(p)}(t) \geq Q\} \leq f\left((Q - n \cdot (p(i)) \cdot L_i - \rho_i \cdot \xi)^+\right) + \frac{1}{(r_i(p) - \rho_i) \cdot \xi} \int_{Q-n(p(i)) \cdot L_i - \rho_i \cdot \xi}^{\infty} f(u) \, du, \quad (10)$$

$$\Pr\{D_i^{(p)}(t) \geq D\} \leq f\left(r_i(p) \left(D - \sum_{l \in p(i)} d_l\right) - n \cdot (p(i)) \cdot L_i - \rho_i \cdot \xi\right)^+ + \frac{1}{(r_i(p) - \rho_i) \cdot \xi} \int_{r_i(p) \left(D - \sum_{l \in p(i)} d_l\right) - n(p(i)) \cdot L_i - \rho_i \cdot \xi}^{\infty} f(u) \, du. \quad (11)$$

□

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 \( f(\sigma) = \Lambda e^{-\alpha \sigma} + B e^{-\beta \sigma} \), i.e., 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 9, in an RPPS PGPS network, for every session \( i \) with SBB traffic and a bounding function \( f(\sigma) = \Lambda e^{-\alpha \sigma} + B e^{-\beta \sigma} \) and for any \( \xi > 0 \) and \( D > 0 \):

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

For the ease of presentation, we omitted the session index \( i \).

Denote the end-to-end delay tail distribution bound, given in (12), as \( \overline{Pr}(D, p) \).

3.3.2.2 Routing Algorithms

We consider the problem of finding a path with the minimal end-to-end delay tail distribution bound. Although the bound (12) is quite complex, it is easy to verify that it is still the same, relatively simple, metric (for a given \( r \)), namely \( \{ \frac{L}{r} + d_l \} \), that characterizes the path with respect to the end-to-end delay tail distribution bound. Thus, an optimal path can be identified through \( K \) executions of Dijkstra’s shortest path algorithm with respect to the metric \( \{ \frac{L}{R_j} + d_l \} \). To achieve lower complexity, one can recur to a rate-quantized, near-optimal solution, such as the following.

**Algorithm Min. delay tail distribution - Rate Quantized (MDD-RQ)**

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 \( \{ \frac{L}{R_j} + d_l \} \) through Dijkstra’s algorithm.
   (d) Compute \( \overline{Pr}(D, p) \).
   (e) \( a_{j+1} \leftarrow \frac{\rho + (1+\varepsilon)}{\rho + (1+\varepsilon) \cdot \frac{(1+\varepsilon)}{2} (R^1 - \rho)} \cdot (R^1 - \rho) \).
   (f) \( R(j + 1) \leftarrow a_{j+1} \cdot R(j) \)
3. Among all paths \( p(j) \), choose a path \( \tilde{p} \) with the minimal end-to-end delay tail distribution bound \( \overline{Pr}(D, p) \).

**Proposition 10.**

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

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

17
2. The algorithm’s complexity is
\[ O \left( (N \log N + M) \cdot \min \left\{ \frac{1}{\varepsilon}, \log \left( \frac{R^K - \rho}{R^i - \rho} \right), K \right\} \right). \]

\[ \square \]

3.3.3 Variable-rate links

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)} \cite{[15]} with parameters \((r, B, \beta)\), if for all \( s, \tau \ (\tau > s > 0) \):

\[ \Pr \left\{ \sum_{n=s+1}^{\tau} S(n) \leq r(\tau - s) - \Delta \right\} \leq B \cdot e^{-\beta \Delta} \]  
(13)

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).

\textbf{Proposition 11.} Let \( I = \{1, 2, \ldots m\} \) be the set of sessions entering an RPPS GPS server with a variable service rate. Let the service rate have exponentially bounded fluctuation with parameters \((R_t, B, \beta)\). For every session \( i \in I \) and for any \( t > 0 \),

\[ \Pr \{ Q_i(t) \geq Q \} \leq \frac{\Lambda_i + B}{1 - e^{-\xi_i(t; r, \rho_i)}} e^{-\xi_i Q}, \]
(14)

\[ \Pr \{ D_i(t) \geq D \} \leq \frac{\Lambda_i + B}{1 - e^{-\xi_i(t; r, \rho_i)}} e^{-\xi_i r_i D}, \]  
(15)

where \( \beta_i \equiv \sum_{j=1}^{m} \frac{\rho_j}{\rho_i} \cdot \beta, \xi_i \equiv \frac{\alpha_i \beta_i}{\alpha_i + \beta_i} \) and \( r_i \equiv \frac{\rho_i}{\sum_{j=1}^{m} \rho_j} \cdot R_t. \]

\[ \square \]

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

\textbf{Proposition 12.} For every session \( i \) with EBB traffic in an RPPS GPS network with EBF service rate, we have:

\[ \Pr \{ Q_i^{(p)}(t) \geq Q \} \leq \frac{\Lambda_i + B^{(p)}}{1 - e^{-\xi_i^{(p)}(t; r_i^{(p)}, \rho_i)}} e^{-\xi_i^{(p)} Q}, \]
(16)

\[ \Pr \{ D_i^{(p)}(t) \geq D \} \leq \frac{\Lambda_i + B^{(p)}}{1 - e^{-\xi_i^{(p)}(t; r_i^{(p)}, \rho_i)}} e^{-\xi_i^{(p)} r_i^{(p)} D}, \]  
(17)

where

\[ \frac{1}{\xi_i^{(p)}} \equiv \frac{1}{\alpha_i} + \sum_{n=1}^{n(p(i))} \frac{1}{\beta_i^{(n)}}, \quad B^{(p)} \equiv \sum_{n=1}^{n(p(i))} B^{(n)} \text{ and } r_i^{(p)} \equiv \min_{n=1, \ldots, n(p(i))} r_i^{(n)} \]

\[ \square \]

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

\textbf{Proposition 13.} For every session \( i \):

\[ \Pr \{ D(p) \geq D \} \leq \frac{\Lambda + \sum_{l \in p} B_l}{1 - e^{\left(1 - \frac{r - \rho}{\pi \sum_{l \in p} \frac{\rho}{\beta_l}} \right) \left( \frac{1}{\pi} \sum_{l \in p} \frac{1}{\beta_l} \right)}} e^{\left(1 - \frac{r - \rho}{\pi \sum_{l \in p} \frac{\rho}{\beta_l}} \right) \left( \frac{1}{\pi} \sum_{l \in p} \frac{1}{\beta_l} \right)}. \]  
(18)

\[ \square \]
3.3.3.1 Routing Algorithms  In [36], we formulate and validate a routing scheme that identifies paths that are efficient with respect to the above bound (18) on the end-to-end delay tail distribution. Since the details are quite lengthy and tedious, they are omitted here.

3.4 Conclusions

The current results contribute to two major subjects within the area of QoS provision. First, they extend the framework of several QoS routing proposals, namely [7, 9, 8], which were designed to operate in conjunction with rate-based service disciplines: while those previous studies exclusively dealt with (deterministically) burstiness constrained (BC) traffic and deterministic guarantees, we investigated the QoS routing problem within the realm of stochastically bounded burstiness (SBB) and stochastic guarantees. Moreover, we also investigated the important case of variable-rate links, both under the deterministic as well as the stochastic settings.

The second contribution of this study was in fact a prerequisite for the first. In order to fully extend the results of [7, 9, 8], 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 and on variable-rate 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, and for BC traffic with 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 [8], e.g., as demonstrated by the non-uniform quantization method applied in algorithm QP-RQ.

To conclude, the aggregate result of the current study is the provision 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.

4 Directions for Future Research

We intend to further investigate QoS provision and routing with end-to-end guarantees. Our focus will be on non-GPS or non-RPPS scheduling disciplines, e.g., CRST GPS and Rate-Controlled scheduling disciplines. Furthermore, we intend to consider statistical end-to-end bounds which exploit statistical multiplexing gain, rather than worst case bounds. In addition, we intend to investigate the process of topology aggregation in the context of information theory. These possible directions are described in the following. Initial results and observations are briefly mentioned and the remaining problems are pointed out.

4.1 Non-RPPS GPS scheduling disciplines

Throughout the analysis, we have focused on RPPS PGPS schedulers. However, many other rate-based scheduling disciplines, which are non-rate-proportional or non-GPS (e.g., SCFQ, STFQ), have been established and explored. Thus, a possible area for future research is the establishment of QoS routing schemes for such disciplines.

In an RPPS GPS scheduling discipline weights are set according to the sessions bandwidth demands. Such a discipline introduces coupling between bandwidth and delay. While simplifying the network management, e.g., call admission control and routing, it leads to a waste of network resources. With non-RPPS disciplines, tighter bounds and, consequently, better network utilization, can be achieved.
For this broader class of GPS assignments, the end-to-end delay bound can be obtained from the universal service curve. The value of the universal service curve at time \( t \), yields a tight bound on the maximum number of session \( i \) bits that can ever traverse the network in the first \( t \) time units of a network session \( i \) busy period. The first slope of the session \( i \) universal service curve, \( U_i \), which is a piecewise-linear function, equals to the session \( i \) long term service rate \( r_i(p) \). Recall that for RPPS GPS networks, the delay bound is inversely proportional to \( r_i(p) \). A tighter bound can be achieved by accounting for the increase in the service rate, which occurs when one of the sessions in the bottleneck ceases to be backlogged. The new service rate is given by the next slope of \( U_i \). Now, the end-to-end delay bound is inversely proportional to two service rates \( r_1 \) and \( r_2 \) (the first and second slopes of \( U_i \)). Accordingly, the following routing sketch, which identifies a feasible path, can be suggested: for each possible value of \( r_1 \) and for each possible value of \( r_2 \) execute a Bellman-Ford shortest path algorithm, with respect to the metric \( \{d_l\} \), in order to identify a shortest path for each possible number of hops \( n \), for all \( 1 \leq n \leq H \). This way, \( O(M^2 \cdot H) \) paths are identified. The solution is found by computing the respective end-to-end delay bound, and choosing the path with the minimal value.

The more general problem, of optimizing the route choice in terms of accommodating multiple connections, is open for future research. The current scheme exploits the multiplexing gain with one impeding session (only the second slope of the universal service curve is considered). One can see that the complexity of such a scheme grows exponentially when more impeding sessions are considered. Obviously, the multiplexing gain and the network utilization increase as more impeding sessions are considered. An interesting direction for future research is the establishment of a polynomial approximation scheme that considers more than one impeding session.

Closed-form bounds for CRST GPS networks under the stochastic settings were established in [6]. These bounds, which are based on the decomposition approach rather than the universal service curve, are more complex. Nevertheless, we would like to further investigate the possibility of establishing simple, yet efficient, routing algorithms corresponding to these bounds.

### 4.2 Non-work-conserving networks

So far we have considered the work-conserving PGPS scheduling discipline. However, the non-work-conserving rate-controlled service disciplines, and in particular the rate-controlled earliest deadline first (RC-EDF), are attractive candidates for providing end-to-end guarantees to individual connections. The main advantages of a rate-controlled service discipline are simplicity of implementation and flexibility. It was also shown in [4] that rate-controlled service disciplines have the additional flexibility of providing end-to-end delay bounds that cannot be guaranteed by the GPS discipline. Furthermore, because of traffic shaping, the network buffer requirements of these disciplines are in general significantly smaller than those of the GPS discipline.

Exact schedulability conditions that detect violations of delay guarantees in a network switch have been established in [29], for both the EDF and SP scheduling disciplines. The corresponding routing problems have been studied in [20, 10, 31]. However, these algorithms do not aim at finding an optimal path, nor an optimal partition of the end-to-end delay bound. Thus, efficient schemes, which aim at reducing the consumption of resources or at maximizing the ability to accommodate future calls, are called for. Obviously, such schemes should efficiently partition the end-to-end delay requirement among the links along the route.

The establishment of an optimal delay allocation scheme is complex due to two main reasons: (i) there does not seem to be a precise optimization criterion, and (ii) such a criterion, which would probably depend on the assigned deadlines (to all channels) on each link along the path, would probably not possess any of the following "nice" properties: continuity, convexity, monotony, etc.

One possible optimization criterion is the available processing time for future channels, which would not violate the requested channel deadline on each link along the path. More precisely, consider the establishment of a new channel \( j \) along a given path \( p \), and suppose that the sum of the MWRT’s along the path is less than the required end-to-end delay; we then seek an allocation scheme that maximizes:

\[
\min_{l \in p} \left( d^l_j - W^l_j \right),
\]
where $d^l_j$ is the assigned deadline of channel $j$ on link $l \in \mathcal{P}$ and $W^l_j$ is the MWRT of channel $j$ on link $l \in \mathcal{P}$. One can see that, on each link $l$, $d^l_j - W^l_j$ is piecewise-linear (w.r.t. $d^l_j$), with no more than $I(l)$ pieces, where $I(l)$ is the number of established channels at link $l$. Accordingly, we believe that the above allocation problem can be solved with complexity of $O(I(l))$, where $I(l) = \max_{l \in \mathcal{P}} I(l)$, and $H$ is the maximal possible number of links along a route. Such a complexity can be achieved by exploiting the piecewise-linearity of the optimization function. Furthermore, we believe that the more general problem, of finding a path that maximizes (19), can be solved with a complexity of $O(I(l) \cdot H \cdot M)$. While being of polynomial complexity, this scheme may be computationally prohibitive, since the maximum number of established channels along the path is typically high. Thus, further investigation is required, in order to establish approximation schemes of lower complexity. Since the suggested optimization criterion is obviously not optimal w.r.t. the ability to accommodate future calls, additional criteria should be proposed and investigated.

Motivated by the statistical bounds on the end-to-end delay established for an EBB traffic model in GPS networks, similar statistical analysis of the rate-controlled, in particular the EDF scheduling discipline, is called for.

Finally, with the schedulability conditions for the SP scheduling discipline at hand, we intend to investigate and establish the corresponding routing schemes.

### 4.3 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 recent works analyzed statistical multiplexing in the context of both the GPS [37, 38] and the EDF [39, 40] scheduling disciplines. A key concept in such an analysis is the effective bandwidth of a session, i.e., is the minimum rate that a source needs to meet its QoS requirements. Here, we would like to investigate previously established statistical end-to-end delay bounds, expand them (where needed) and establish the corresponding routing schemes.

### 4.4 Hierarchical networks and topology aggregation

PNNI routing, designed for ATM networks, has an hierarchical structure [33]. In hierarchical routing, nodes and links at an 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.

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 [36] we consider a general model, suitable for hierarchical networks, where each link is composed of one or several hops. There, we show that the routing algorithms proposed in this study with slight modifications can be efficiently applied in hierarchical networks.

Since the actual number of hops is not always known to the source, an estimation of this value is desirable, e.g., for routing purposes. A reasonable assumption is that the link delay and bandwidth are highly correlated with its number of hops. Thus, the number of hops can be estimated from these values. We have initial result in this respect; specifically, we established an estimation of the number of hops when the link bandwidth and delay are assumed to be independent and exponentially distributed. Obviously, these assumptions are quite naive. Further investigation is required in order to efficiently estimate the number of hops under more realistic assumptions.
Consider now the establishment of topology aggregation schemes. Here, we propose the employment of rate-quantization for joint aggregation of link delay and bandwidth. Most related work on topology aggregation considers the aggregation of each metric separately. However, since the proposed QoS routing algorithms in this study, 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.

Recall that topology aggregation is the process of summarizing and compressing topology information. We intend to exploit knowledge on information theory in order to establish efficient compression schemes. Accordingly, we consider topology aggregation as the process of graph compression. In [41], 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 LZW tree in LZW compression is used to compress a binary tree generated by a stationary ergodic source. Encouraged by this work, we intend to 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 [41], considers only the structure of the graph/tree, it 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.
References


