Università di Pisa
Dipartimento di Informatica

Technical Report

Delay-constrained Routing Problems: Accurate Scheduling Models and Admission Control

Antonio Frangioni Laura Galli Giovanni Stea

December 29, 2015

address: Largo B. Pontecorvo 3, 56127 Pisa, Italy. tel: +39 050 2212700 fax: +39 050 2212726

Delay-constrained Routing Problems: Accurate Scheduling Models and Admission Control

Antonio Frangioni * Laura Galli * Giovanni Stea †

December 29, 2015

Abstract

It has been recently shown [7] that the problem of routing a new packet flow in a computer network subject to a constraint on the worst-case end-to-end delay of its packets can be formulated as a Mixed-Integer Second-Order Cone Program (MI-SOCP), and solved with general-purpose tools on instances of realistic size in time compatible with real-time use. However, that result was obtained for only one of the classes of schedulers in use in today's networks, namely *Strictly Rate-Proportional* ones. Furthermore, that result relies on assuming that each link is "fully loaded", so that the *reserved rate* of the flow coincides with its *guaranteed rate*. These assumptions entail both simple latency expressions, and the fact that flows are *isolated* from each other as far as their end-to-end delay is concerned; that is, admitting a new flow does not increase the end-to-end delay of the existing ones. However, other classes of scheduling algorithms commonly found in current networks both yield more complex latency formulæ and do not enforce strict flow independence. Furthermore, the delay actually depends on the *guaranteed* rate of the flow, which can be significantly larger than the *reserved* rate if the network is less loaded. In this paper we extend the result to other classes of schedulers and to a more accurate representation of the latency, showing that, despite the increase in complexity, the problem is still efficiently solvable, even when admission control needs to be factored in, for realistic instances, provided that the right modeling choices are made.

Keywords: *Routing problems, maximum delay constraints, scheduling algorithms, admission control, Mixed-Integer NonLinear Programming, Second-Order Cone Programs, Perspective Reformulation*

*Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy. E-mail: {frangio, galli}@di.unipi.it

†Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Largo Lucio Lazzarino 1, 56122 Pisa, Italy. E-mail: giovanni.stea@unipi.it

1 Introduction

Current computer networks are almost exclusively packed-based, and their fundamental protocols were designed to ensure *reliable* delivery of information. That is, every effort is made to ensure that all packets—possibly after having been retransmitted several times—eventually reach their destination, with little heed about the time it takes to do so. However, the Internet has evolved far beyond its initial design goals, and now has to support applications that require stringent guarantees on end-to-end delays (voice/video streaming, remote operation of industrial/medical tools, etc.). This means that obtaining guarantees on the *Quality of Service* (QoS) of a given *packet flow*, such as a minimum guaranteed rate and a maximum delay, is now crucial; however, doing so in a fundamentally packed-based infrastructure is highly nontrivial.

A basic problem regarding QoS in computer networks concerns deciding whether or not a new packet flow (hereafter, just "flow") can be admitted in a network and obtain a required QoS with the resources currently left free from the existing ones, i.e., without disrupting them (making them to violate their QoS requirements). In this case, one typically wants to minimize the amount of network resources used by the newly admitted flow, so as to leave as much capacity as possible available to future flow requests. In general, these *QoS routing problems* can be formulated as Constrained Shortest Path (CSP) problems. Some of these CSP problems admit polynomial algorithms, such as those where a single end-to-end constraint is an additive or multiplicative *concave* function of per-link metrics; this is why, in the literature, packet delay is usually considered additive (i.e., a fixed delay for each link), and per-link loss probabilities are considered multiplicative (i.e., independent). However, as could be expected, CSP with two or more constraints, even additive or multiplicative ones, is already $\mathcal{NP}$-complete [30, 20]. Due to the typical strict requirements on the time to deliver the solution in practice (say, some 10s or 100s of milliseconds), it is natural that several efforts have been made to develop approximate approaches for the corresponding CSP (e.g., [15, 33, 16, 13]). As an alternative, stochastic traffic models are used to compute guarantees on *average* QoS metrics [25]. These are however unrelated with worst-case guarantees.

All the above references rely on rather simplified network models where the relevant QoS parameters, say link delays, are considered statically known and additive. This neglects *queueing*, i.e., the link delay due to the fact that the same link is shared by different flows, which may force some flow's packets to wait until other flow's packets have been processed. Queueing delays are intimately dependent on the details of the *packet schedulers* employed to arbitrate between different flows; for instance, it is easy to realize that the obvious FIFO scheduler does not guarantee any fairness on resource allocation, and therefore on the Worst-case Delay (WCD) that a packet can suffer while traversing a link. This has motivated a consistent stream of research on schedulers that can guarantee that packets meet their deadline on a link under certain conditions. The cornerstone of the packet scheduling literature is the Generalized Processor Sharing (GPS) [24], a paradigm that defines an ideal reference system which serves backlogged flows simultaneously at a rate proportional to their *weight*. GPS schedulers can provide a minimum guaranteed rate for each flow under certain conditions: for instance, if the weight of a flow is chosen equal to its *reserved rate*, then the *guaranteed rate* it gets is at least as large as the reserved one, as long as the sum of the reserved rates does not exceed the link capacity. This, in turn, allows per-link, and ultimately

end-to-end, WCD bounds to be computed if the arrival rate at the source is constrained. Two practical implementations of GPS have been proposed, namely *Packet-by-packet Generalized Processor Sharing* (sometimes also called Weighted Fair Queueing) [24] and *Worst-case Fair Weighted Fair Queueing* [4]. Both exhibit tight guarantees on the *latency*, i.e., the worst-case scheduling delay at a link; in particular, their latency is—barring a small additive constant—inversely proportional to the guaranteed rate, thereby earning them the moniker of *Strictly Rate-Proportional* (SRP) schedulers.

When SRP schedulers are employed, it is possible to devise QoS routing problems that exploit the relationship between the maximum end-to-end delay and the guaranteed rates for a flow at each link along its path. Within this stream of literature, [22, 23] show that the problem of finding a path with a pre-specified maximum end-to-end delay is $\mathcal{NP}$-hard in general, unless the *same rate* is allocated at each link. However, allocating the same rate at each link may be too constraining [21, 6], i.e., results in much larger reserved rates (and, therefore, lower overall network performance) than would be possible if non-uniform rate allocation was allowed. Recently, [7] presented exact, approximate and heuristic approaches to that latter problem, showing that—despite $\mathcal{NP}$-hardness—optimal solutions can be found in split-second times for realistic-sized networks. Furthermore, it has been shown in [8], by means of extensive simulations on real-world networks with realistic data, that joint path computation and non-uniform rate allocation leads to remarkable performance gains in terms of flow blocking probability, significantly outperforming simpler routing approaches (e.g. route-first-allocate-second, or those based on uniform rate allocation) while still being solvable in a running time compatible with actual on-line network management operations.

Unfortunately, SRP schedulers require a relatively high number of operations to select a packet for transmission, which is a downside on high-speed links and/or with many simultaneous flows. In the last two decades, a relevant amount of literature has therefore been devoted to devising other schedulers which exhibit a different trade-off between latency and implementation cost. Some of these schedulers have also made their way into commercial hardware, cf. [19] among others. At one end of the spectrum we find approximations of the GPS paradigm based on flow grouping [5, 14], and at the other end lie frame-based algorithms such as Deficit Round Robin [27] and its derivatives [17, 18, 19]. In both cases, latency guarantees are looser than those of SRP schedulers, but their implementation cost is smaller as well. To the best of our knowledge, no attempt has been made so far to devise QoS-routing schemes for these other schedulers. The only related work that we are aware of, [21], shows that non-uniform rate allocation *given a pre-specified routing plan* achieves better network utilization than uniform rate allocation in the presence of end-to-end delay constraints. This means that so far it has been impossible to estimate the impact of employing lower-complexity schedulers on the network performance (e.g., utilization or failure probability).

Furthermore, all previous works—comprised [7, 8]—have resorted to simplifying the latency formulæ by assuming that the *guaranteed* rate of a flow is equal to its *reserved* rate. The reserved rate in fact lower bounds the guaranteed rate, and the two are only equal if all the links of the path are used up to capacity; in less loaded scenarios, the two may differ greatly. We refer to the latter hypothesis as the *bound assumption* hereafter. Hence, so far QoS routing has been performed using over-estimates of the WCD experienced by a flow,

which leads to a more conservative resource allocation than would be necessary. To the best of our knowledge, the impact of the bound assumption on the network performance has not been investigated yet.

This paper provides a first, necessary step towards answering the above questions by formulating and solving the *Admissible Delay-Constrained Shortest Path* (ADCSP) problem: given the current state of the network, a set of link *reservation costs*, and a new flow to be routed between a given source and destination under a pre-specified end-to-end delay constraint, determine a feasible path and a feasible rate reservation on each link (if any exists) minimizing the total reservation cost and ensuring that existing flows still satisfy their end-to-end delay constraints. We show that, for several classes of packet schedulers, the ADCSP problem can be formulated as a Mixed-Integer Second-Order Cone Problem (MI-SOCP) and solved by general-purpose tools in split-second times for instances corresponding to realistically-sized networks. This paves the way to exploring the impact of employing different scheduling algorithms on network performance. We also show that, while distinguishing between *reserved* and *guaranteed* rates in the latency formulæ does increase the complexity of the models, the cost of doing so remains bearable, thus opening the way to studying the impact of these modeling choices, too, on network performance.

The paper is organized as follows: in Section 2 we describe the model of the communication network that we employ, we review the latency formulæ of the different schedulers, and we recall the corresponding end-to-end flow delay expressions. In Section 3 we recall the MI-SOCP model proposed in [7] and used in [8] only for the case of SRP schedulers and under the bound assumption; we then improve it, and extend it to several other well-known schedulers. To do so, we need to introduce explicit *admission control constraints* that ensure that the existing flows still meet their deadline without changing their path and reserved rates; these constraints were implicit for SRP (under the bound assumption), but are not so (whether or not that assumption is maintained) for all other schedulers. In Section 4 we extend the models using guaranteed rates, as opposed to reserved rates, in the *link latency* expressions alone. In Section 5 we deal with the effect of using guaranteed rates in the *end-to-end delay* expressions. Finally, in Section 6 we report computational results which show the relative efficiency of the various models on real networks with realistic traffic data, and in Section 7 we draw some conclusions.

2 System model

We are given a computer network represented by a directed graph $G = (N, A)$. Nodes are switching elements (e.g., routers), and arcs are the link interconnecting them. Our problem is to route a single "new" flow along a single path through the network, from a given origin $s \in N$ to a given a destination $d \in N \setminus \{s\}$, allocating the minimum possible amount of resources (reserved rates on arcs) so that the *Worst-Case Delay* (WCD)—the maximum end-to-end delay that any packet may incur during the traversal—is below a given *deadline* δ. We assume our flow to be characterized by an *arrival curve* $A(\tau) : \mathbb{R}_+ \to \mathbb{R}_+$ specifying how many bits of that flow are allowed to be injected at the origin s in any interval of duration τ. In other words, if the *cumulative arrival function* $F(t)$ measures how many bits have entered the origin at time t, we have $F(t + \tau) - F(t) \leq A(\tau)$ for all t and $\tau \geq 0$. For our purposes

we assume the arrival curve to be entirely specified by the two parameters σ (burst) and ρ (rate) of a *leaky-bucket traffic shaper*, so that $A(\tau) = \sigma + \rho \cdot \tau$. Each arc $(i, j) \in A$ is characterized by a fixed *link delay* l_{ij}, a *physical link speed* w_{ij}, and a *reservable capacity* c_{ij} ($\leq w_{ij}$, since in general a network administrator may want to set some capacity apart for other purposes, e.g., control traffic). Each node $i \in N$ in the network is characterized by a fixed *node delay* n_i; also, the *maximum transmit unit* (MTU) L is known and assumed to be constant for simplicity.

The aim of this paper is to discuss mathematical models for the *Admissible Delay-Constrained Shortest Path* (ADCSP) problem: given the current state of the network and a set of link *reservation costs* f_{ij} (the cost of reserving one unit of capacity on (i, j)), find one feasible *s-d* path p—if any exists—and a feasible reservation of capacity for each of its arcs so that the flow can be routed along p with the given reserved capacities, and *both the new flow and all the existing ones* meet their deadline, at the minimum possible reservation cost for the new flow.

Of course, ADCSP requires to compute the WCD of a flow. This depends on several factors:

1. The selected *routing* for the flow, i.e., the *s-d* path p in G.

2. For each arc $(i, j) \in p$, the *reserved rate* $0 \leq r_{ij}$ ($\leq c_{ij} \leq w_{ij}$) for the flow along the arc.

3. The specific characteristics of the schedulers employed to share the output links' bandwidth among the flows, notably their latency guarantees. For the sake of simplicity we will always assume the schedulers to be the same at each link, but extending the models to non-uniform cases is obvious.

4. Depending on the previous point, the paths and/or reserved rates of all the other flows currently active in the network. In particular, in the following we will denote by $P(i, j)$ the set of existing paths (*not* counting the one just to be routed) traversing arc (i, j), and by r_{ij}^q the corresponding reserved rate of each flow $q \in P(i, j)$. We will also find it expedient to consider A partitioned into $A' \cup A''$, where A' contains the arcs (i, j) that are "empty" ($P(i, j) = \emptyset$) and A'' those that contain at least one flow.

While the natural decision variables of the problem are the reserved rates r_{ij} at each link (cf. point 2 above), in general the WCD rather depends on the *guaranteed rate* obtained by the flow. For all the fair-queueing schedulers that we will examine, the guaranteed rate is at least as large as the reserved rate. In particular, assuming that the total reserved rate r_{ij}^{tot} along an arc for all flows (*comprised* the newly routed one) satisfies $r_{ij}^{tot} \leq w_{ij}$, all these schedulers ensure that

$$g_{ij} = r_{ij}(w_{ij}/r_{ij}^{tot}) \geq r_{ij} \tag{1}$$

(whence the need of knowing the r_{ij}^q, cf. point 4 above). Since $r_{ij}^{tot} \geq r_{ij}$ (the rate of the current flow is counted together with all the others), it is easy to see that $g_{ij} = w_{ij}$ when the arc is "completely unloaded" (the current flow is the only one having reserved on it, i.e., $r_{ij}^{tot} = r_{ij}$, which implies $(i, j) \in A'$), while $g_{ij} = r_{ij}$ when the arc is "completely loaded", i.e.,

$r_{ij}^{tot} = w_{ij}$. We will therefore always ensure that

$$r_{ij} \leq \bar{c}_{ij} = c_{ij} - \bar{r}_{ij} \ (\leq c_{ij} \leq w_{ij}), \quad \text{where} \quad \bar{r}_{ij} = \textstyle\sum_{q \in P(i,j)} r_{ij}^{q} \ , \tag{2}$$

in order to be able to write the guaranteed rate as a function of the reserved one:

$$g_{ij} = (w_{ij} r_{ij}) / (\bar{r}_{ij} + r_{ij}) \ . \tag{3}$$

In order for the WCD to be *finite*, the minimum *guaranteed* rate among all links of the path must be at least as large as the traffic injection rate of the flow, i.e.,

$$g_{ij} \geq \rho \qquad \forall (i,j) \in p \ . \tag{4}$$

If (4) is satisfied, the general form of the WCD for a given routing path p is

$$\frac{\sigma}{\min\{ \, g_{ij} \, : \, (i,j) \in p \}} + \sum_{(i,j) \in p} \left(\, \theta_{ij} + l_{ij} + n_i \, \right) \ , \tag{5}$$

where θ_{ij} is the *link latency* experienced by the flow on path p when traversing the arc (i,j), i.e., the delay that a packet may undergo due to the presence of competing flows. Note that θ_{ij} does not incorporate fixed delays, which in fact are separately considered in (5).

The exact form of θ_{ij} depends on the scheduling algorithm employed at the link. The simplest expression, henceforth called Strictly Rate-Proportional (SRP) latency, is the one of Packet-by-packet GPS [24] and Worst-case Fair Weighted Fair Queueing [4]. The exact expression for SRP latency is

$$\theta_{ij} = \frac{L}{w_{ij}} + \begin{cases} L/g_{ij} & \text{if } (i,j) \in A'' \\ 0 & \text{otherwise} \end{cases} \ . \tag{6}$$

In all the previous developments in the literature [22, 23, 21, 7, 8], a "pessimistic" view has been taken where one assumes that

$$g_{ij} = r_{ij} \qquad \forall (i,j) \in p \ ; \tag{7}$$

in turn, this can be taken to logically imply that $(i,j) \in A''$. All this yields the simplified link latency formula

$$\theta_{ij} = \frac{L}{r_{ij}} + \frac{L}{w_{ij}} \ . \tag{8}$$

We will refer to (8), and to all the other delay formulaæ obtained under the bound assumption (7), as the *bound* delay estimates, as opposed to the the more accurate *worst-case* estimates obtained by eschewing the simplifying assumption and employing the actual formulæ based on the guaranteed rates, such as (3). Of course, bound estimates on the WCD are safe; however, they may be arbitrary loose. The simplified formula (8) more easily motivates the SRP moniker, since the link latency (which makes for a significant part of the end-to-end delay, as per (5)) is almost strictly proportional to the (inverse of) the reserved rate r_{ij}; the constant offset L/w_{ij}, which may appear to thwart "strict" rate/latency proportionality, is due to the atomic transmission of packets that cannot be avoided in any real system.

Beside being simple, (8) can be appropriately rewritten in such a way that ADCSP can be formulated as a Mixed-Integer Second-Order Cone Program (MI-SOCP), as shown in [7] and later on (cf. (16)–(23)). Furthermore, (8) implies that the latency of a flow is not influenced by the path and/or reserved rates of the existing flows. It is easy to see, as discussed in detail below, that under (7) the same holds for the overall end-to-end delay formula. This means that checking whether or not a new flow can be routed in the network amounts to finding a delay-constrained path for that flow *using only the residual rate $\bar{c}_{ij}$ left by the other ones*, with no danger that any other flow violates its deadline. In other words, under the bound assumption and SRP latency, *admission control only boils down to the constraints $r_{ij} \leq \bar{c}_{ij}$*. It is not surprising, then, that this setting has been the only one explored so far.

Unfortunately, SRP latency can only be achieved by schedulers that exhibit at least $O(\log n)$ worst-case per-packet complexity [32, 28], n being the number of flows traversing the link[1]. Such complexity is often considered a burden, especially at high link speeds—when only few *ns* are available to make scheduling decisions—and/or when a large number of flows are to be scheduled. For this reason, the literature on packet scheduling of the last two decades abounds with proposals of schedulers which trade an increase in latency for a lower complexity. Some of the proposed solutions have made their way into commercial or however widespread implementations (i.e., network routers and switches [19], or NIC drivers for open-source systems [5]). While fair-queueing schedulers may differ regarding several properties, from the point of view of latency a large number of them falls into one of the following categories:

- *Group-based approximations of SRP*, e.g. [5, 14]. These schedulers group flows are according to their reserved rate, at logarithmic intervals. This, and the usage of clever data structures, imply that a scheduling decision requires $O(1)$ complexity. However, this comes with an increase in the latency. The "bound" link latency expression is [5]

$$\theta_{ij} = 3\frac{2^{\lceil \log_2 w_{ij}L/r_{ij} \rceil}}{w_{ij}} + 2\frac{L}{w_{ij}} \quad , \tag{9}$$

  which can be easily shown to satisfy

$$3\frac{L}{r_{ij}} + 2\frac{L}{w_{ij}} \leq \theta_{ij} \leq 6\frac{L}{r_{ij}} + 2\frac{L}{w_{ij}} \quad . \tag{10}$$

  Hence, the latency is still (approximately) rate-proportional, but between 3 and 6 times larger than (8) (disregarding the constant term, which is twice as large). We call (9) a *Group-based Strictly Rate-Proportional* (GSRP) latency. Interestingly, to the best of our knowledge [29] "worst-case" versions of the link latency formulæ of Group-based approximations not only are not know, but are considered to be unlikely to exist at all. In other words, for these schedulers only reserved rates, as opposed to guaranteed rates, matter. This means that for this particular group of schedulers, only the "bound" version of the link latency formulæ can be used.

[1]Note that the complexity of WFQ and WF^{2}Q was believed to be $O(n)$ until 2004, i.e., more than ten years after the publication of [24], when its *exact* logarithmic-cost implementation was first described in [28]. This also justifies the existence of many works in the literature trying to *approximate* GPS at a lower complexity, e.g. $O(\log n)$, to which we will make reference in the rest of the paper.

- Schedulers with *Weakly Rate-Proportional* (WRP) latency, e.g., Self-Clocked Fair Queueing [12]. These still exhibit logarithmic complexity, which was considered an advantage at the time of their proposal given that SRP schedulers were considered to be $O(n)$, but their latency depends on the *number of flows* $|P(i,j)|$ traversing the link simultaneously:

$$\theta_{ij} = |P(i,j)|\frac{L}{w_{ij}} + \frac{L}{g_{ij}} \quad . \tag{11}$$

  The bound version—under (7)—of the above worst-case formula is therefore

$$\theta_{ij} = |P(i,j)|\frac{L}{w_{ij}} + \frac{L}{r_{ij}} \quad . \tag{12}$$

  The adjective "weakly" rate-proportional comes from the fact that the non-rate-proportional offset in (11), (12) is larger than SRP's, especially if $|P(i,j)|$ is high. Thus, increasing the reserved rate may decrease the latency only marginally. As we shall see, in this case even the bound assumption (7), which yields the simplified expression (12), is not enough to isolate flows from each other, precisely due to the term depending on $|P(i,j)|$.

- *Frame-based* (FB) schedulers, such as Deficit Round Robin [27] and similar [17, 18], avoid emulating a GPS server altogether, and impose that flows are visited in a fixed order, each for a minimum amount of time called a *quantum*. The guaranteed rate is thus the ratio of the quantum to the round duration. Thus, in these schedulers the latency depends on the number of flows, but also on their quantum, hence on the reserved rate for *each* flow. FB schedulers only achieve $O(1)$ complexity if all quanta are *lower bounded*; with DRR, such lower bound is the MTU L. However, the ratio of the reserved (and guaranteed) rates between any two flows matches that of their quanta: thus, the flow requesting the minimum reserved rate must get a quantum equal to the lower bound L, and all other flows get their quanta scaled accordingly. All in all, the latency expression of DRR, besides the number of flows, also depends on the reserved rates of other flows on the link; not only their sum $\bar{r}_{ij}$, but also their minimum:

$$r_{ij}^{min} = \min\{\, r_{ij}^{q} \,:\, q \in P(i,j) \,\} \;\; (\geq \rho).$$

  Some straightforward algebraic manipulations on the expression reported in [18] give

$$\begin{aligned}
\theta_{ij} &= \frac{L}{w_{ij}}\left[\bar{r}_{ij}\left(\frac{1}{\min\{\, r_{ij}\,,\, r_{ij}^{min}\,\}} + \frac{1}{r_{ij}}\right) + |P(i,j)| + 1\right] \\
&= \frac{L}{w_{ij}}\left[\frac{\bar{r}_{ij}}{\min\{\, r_{ij}\,,\, r_{ij}^{min}\,\}} + \left(\frac{\bar{r}_{ij}}{r_{ij}} + 1\right) + |P(i,j)|\right] \\
&= \frac{L}{w_{ij}}\frac{\bar{r}_{ij}}{\min\{\, r_{ij}\,,\, r_{ij}^{min}\,\}} + |P(i,j)|\frac{L}{w_{ij}} + \frac{L}{g_{ij}} \quad .
\end{aligned} \tag{13}$$

  Note that in the second step we have used (3) to rewrite $(1/w_{ij})(\bar{r}_{ij}/r_{ij} + 1)$ as $1/g_{ij}$. Also, note that the first and second terms are null when $(i,j) \in A'$, i.e., there are no previous flows traversing the arc ($\Longrightarrow |P(i,j)| = 0$ and $\bar{r}_{ij} = 0$). The bound assumption

(7), besides $g_{ij} = r_{ij}$, also implies that $\bar{r}_{ij} + r_{ij} = w_{ij}$; this yields the slightly simpler formula

$$\theta_{ij} = \frac{L}{w_{ij}} \frac{w_{ij} - r_{ij}}{\min\{\, r_{ij} \,,\, r_{ij}^{min} \,\}} + |P(i,j)| \frac{L}{w_{ij}} + \frac{L}{r_{ij}} \quad . \tag{14}$$

Still in the FB class other schedulers can be found (e.g. [17, 18]) which improve on the basic DRR scheme by allowing quanta to be scaled down by a constant factor κ, while still retaining $O(1)$ complexity by leveraging clever data structures. As a consequence, their latency is smaller, and it has the following expression, which is similar to the previous one:

$$\begin{aligned} \theta_{ij} &= \frac{L}{w_{ij}} \left[\bar{r}_{ij} \left(\frac{1}{\kappa \cdot \min\{\, r_{ij} \,,\, r_{ij}^{min} \,\}} + \frac{1}{r_{ij}} \right) + |P(i,j)| + 1 \right] \\ &= \frac{L}{w_{ij}} \frac{\bar{r}_{ij}}{\kappa \cdot \min\{\, r_{ij} \,,\, r_{ij}^{min} \,\}} + |P(i,j)| \frac{L}{w_{ij}} + \frac{L}{g_{ij}} \quad . \end{aligned} \tag{15}$$

When $\kappa \to \infty$, (15) approaches the WRP latency (11); however, κ also influences the implementation cost, hence cannot be arbitrarily large. It is easy to see that there is a diminishing return in increasing κ; we refer the interested reader to [18] for details. For our purposes, κ is only a multiplicative constant in the formula and has no impact on the shape of the optimization models; hence, for simplicity in the following we will only work with $\kappa = 1$, i.e., (13)/(14).

Comparing the "bound" formulæ (8) and (12) shows that SRP schedulers have smaller latency than WRP ones. There would seem to be an exception for the case of the "completely unloaded" arcs in A'. However, looking at the corresponding worst-case formulæ (6) and (11) shows that the exception is only a figment due to the overestimate: for $(i,j) \in A'$, both schedulers give the same L/w_{ij} latency, which is clearly the minimum possible (a packet needs to be fully received before it can be re-transmitted). Comparing (12) with (14) also shows that, cœteris paribus, the WRP latency is always smaller than or equal to that of FB; this is confirmed by the worst-case versions (11) and (13).

Note that, in general, the link latency expression of all schedulers depends on the routing and bandwidth choices for *all* the other flows; the only exception is SRP under the bound assumption (7). In other words, routing one new flow p in the network can (and does) impact on the delay of any existing flow q that shares with p at least one arc (i,j), for several different reasons:

- $\bar{r}_{ij}$ as "perceived" by q increases, which means that the guaranteed rate of q decreases;
- for WRP and FB, the term $|P(i,j)|$ increases for all arcs of p;
- for FB only, if the reserved rate r_{ij} of the new flow is strictly smaller than r_{ij}^{min}, then the delay caused to q by that arc will also increase.

All this means that if the routing of the new flow is only possible under the conditions that the routing/rates decisions for existing flows cannot be changed and that they remain delay-feasible, further, non obvious *admission control* mechanisms must be put in place. To

the best of our knowledge, no work on QoS routing so far has investigated incorporating global-scale admission control tests in the routing algorithm. The reason is that all previous work on the subject has focused on SRP schedulers under (7), that have a very peculiar characteristic: since the corresponding link delay formula (8) does not contain g_{ij}, $\bar{r}_{ij}$, r_{ij}^{min} and $|P(i,j)|$, it actually does not depend on the state of the network, except for the requirement (2) on the capacities. Hence, assuming that the reserved rate satisfies (2) is all that is needed to implement admission control. Note that the same also holds for the Group-based approximations.

In the following we extend the set of delay-constrained routing models available in the literature, in three different ways:

1. by considering not only (G)SRP schedulers but also WRP and FB ones;
2. by considering the effect of removing the bound assumption (7) in both the arc latency and the overall end-to-end delay formulæ;
3. by including explicit provisions for access control whenever they are needed (i.e., everywhere except for the very special case of SRP under (7)).

In the next section we will first review the (A)DCSP models proposed in [7] for the case of (8) under the bound assumption (7), and then discuss the issues related to extending them to the other scheduling algorithms. As we shall see, this will require to explicitly model admission control.

3 Admissible Delay-Constrained Shortest Path

Following [7], we start by modeling the ADCSP using the *bound* formulæ. We will later discuss what modifications are required to insert the guaranteed rates in place of the reserved ones.

A natural way to construct a Mixed-Integer NonLinear model of ADCSP is to introduce arc-flow variables $x_{ij} \in \{0,1\}$ indicating whether or not arc (i,j) belongs to the path p chosen for the "new" flow, together with variables r_{ij} indicating the amount of reserved rate on (i,j); clearly, $r_{ij} = 0$ if $x_{ij} = 0$. Then, we can define the following set of constraints that can be a part of an ADCSP formulation irrespectively of the choice of the scheduling

algorithm:

$$\min \textstyle\sum_{(i,j)\in A} f_{ij} r_{ij} \tag{16}$$

$$\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = \begin{cases} -1 & \text{if } i = s \\ 1 & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \qquad i \in N \tag{17}$$

$$\sigma t + \textstyle\sum_{(i,j)\in A} [\, \theta_{ij} + (l_{ij} + n_i) x_{ij} \,] \le \delta \tag{18}$$

$$t\, r_{min} \ge 1 \;\;,\;\; t \ge 0 \tag{19}$$

$$r_{ij} \le \bar{c}_{ij} x_{ij} \qquad (i,j) \in A \tag{20}$$

$$\rho x_{ij} \le r_{ij} \qquad (i,j) \in A \tag{21}$$

$$\rho \le r_{min} \le r_{ij} + \bar{c}_{max}(1 - x_{ij}) \qquad (i,j) \in A \tag{22}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A \tag{23}$$

The objective function (16) represents the total reservation cost; most often $f_{ij} = 1$, which can be useful for algorithmic purposes [7]. Note that the problem could be generalized in a trivial way to the case where the cost is any non-negative combination of reservation cost and a given fixed cost for selecting any arc. The standard flow conservation constraints (17) ensure that the x_{ij} variables represent a *s-d* path. Constraint (18) imposes the end-to-end delay restriction, implementing (5) provided that appropriate conditions are imposed so that the variables θ_{ij} correctly represent the link latencies; this will be discussed in the following separately for each scheduler. Note that the auxiliary variable t in (18) together with the (rotated) Second-Order Cone constraint (19) represents the term σ/r_{min} in (5); thus, the above is a fragment of a Mixed-Integer SOCP. As we will see, all the delay formulæ can be represented by SOCP constraints, so that all our formulations can be tackled by means of general-purpose MI-SOCP solvers. Constraints (20)–(22) enforce the semi-continuous nature of reserve rate variables r_{ij}, i.e., $x_{ij} = 0 \Longrightarrow r_{ij} = 0$ and $x_{ij} = 1 \Longrightarrow r_{ij} \in [\rho, c_{ij}]$. Note in particular that (22) both enforces (4) (under (7)) and ensures that $r_{min} \le r_{ij}$ when $x_{ij} = 1$, while the use of the "big-M" $\bar{c}_{max} = \max\{\, \bar{c}_{ij} \,:\, (i,j) \in A \,\}$ ensures that any arc not in the chosen path ($x_{ij} = 0$) does not contribute to setting r_{min}.

The above model has been devised for the case of (8). There, the main issue is to express

$$\theta_{ij} = \begin{cases} \frac{L}{r_{ij}} + \frac{L}{w_{ij}} & \text{if } x_{ij} = 1 \\ 0 & \text{if } x_{ij} = 0 \end{cases} \;\;,$$

which is made slightly nontrivial by the fact that $x_{ij} = 0 \Longrightarrow r_{ij} = 0$, thereby rendering the L/r_{ij} term ill-defined. This requires appropriate formulation techniques to be obtained within the limitations of a MI-SOCP problem; in [7] it is proven that the best approach makes use of *Perspective Reformulation techniques* [10, 11], and results in

$$\theta_{ij} = L s_{ij} + (L/w_{ij}) x_{ij} \qquad (i,j) \in A \tag{24}$$

$$s_{ij} r_{ij} \ge x_{ij}^2 \;\;,\;\; s_{ij} \ge 0 \qquad (i,j) \in A \tag{25}$$

Clearly, the "$(L/w_{ij})x_{ij}$" term in (24) can be merged into the corresponding "$(l_{ij} + n_i)x_{ij}$" term in (18), which means that we only have to implement it as

$$\sigma t + \sum_{(i,j)\in A} \left[\, L s_{ij} + (L/w_{ij} + l_{ij} + n_i)x_{ij} \,\right] \leq \delta \tag{26}$$

subject to (25). Because the second term in the sum in (26) will be common to many cases, for the sake of notational simplicity we define $\bar{l}_{ij} = L/w_{ij} + l_{ij} + n_i$. Similarly to (19), the rotated SOCP constraint (25) ensures that $s_{ij} \geq 1/r_{ij}$ when $x_{ij} = 1$, while it allows $s_{ij} = 0$ (and hence $\theta_{ij} = 0$) when $x_{ij} = 0$. Let us remark here that in [7] the conic constraint in (25) is scaled with L, i.e., it reads

$$s_{ij} r_{ij} \geq L x_{ij}^2 \ ;$$

clearly, then one has a term "s_{ij}" rather than "Ls_{ij}" in (26). While this is basically equivalent for the SRP model, the "unscaled" version (24)–(25) will be shown later on to be useful in several cases.

The formulation (16)–(23), completed with (24)–(25), can be solved in various ways, the simplest one being passing it to a general-purpose MI-SOCP solver like `Cplex` or `Gurobi`. A judicious combination of this and combinatorial heuristics has been shown in [7] to be efficient and effective for solving realistic instances. In the following we will discuss how to extend it to other schedulers, still under the bound assumption (7). However, we start with discussing some—to the best of our knowledge, new—reformulations of some fragments of the model. These have both an interest in themselves, as they may make the solution process more efficient, and are relevant for the discussion in §5.

3.1 Reformulating the delay constraint

The simple observation at the basis of this paragraph is that expressing the term "$1/\min\{\, r_{ij} : (i,j) \in p \,\}$" in (5) via the variable t in (18), linked to r_{min} (and hence the r_{ij}) via the SOCP constraint (19), is completely general and does not depend on the specific form of the expression for θ_{ij}. However, *all link latency formulæ we will work with contain at least a term proportional to* $1/r_{ij}$. This is necessarily implemented, in a MI-SOCP model, via the extra variable s_{ij} subject to (25). This paves the way to rewriting that term as

$$t = \max\{\, s_{ij} = 1/r_{ij} \,:\, (i,j) \in p \,\} = 1/r_{min} \ . \tag{27}$$

In other words, constraints (19) and (22) could in principle be replaced with the simpler

$$t \geq s_{ij} \qquad (i,j) \in A \tag{28}$$

eliminating the variable r_{min}. This, on the outset, looks advantageous. On one hand it avoids one conic constraint and one variable. On the other hand, it avoids the use of the "big-M" $\bar{c}_{max}$ in (22), since arcs for which $x_{ij} = 0 \Longrightarrow r_{ij} = 0 \Longrightarrow s_{ij} = 0$ are naturally irrelevant to define t.

However, it is easy to prove that neither formulation dominates the other. Indeed, consider the elementary instance on a graph with two nodes and two identical parallel arcs between them, where $\rho = f = \bar{l} = 1$, $\sigma = L/w = \bar{c} = \bar{c}_{max} = 10$, and $\delta = 3$. One can

numerically verify that the optimal solution to the continuous relaxation of (16)–(25) has $x^* = 1/2$, $r^* = 5$, $s^* = 0.05$ (identical on both arcs for obvious symmetry reasons), $r^*_{min} = 10$ and $t^* = 0.1$, for an objective function value of 10. Incidentally, in this case the continuous relaxation is tight, because the optimal integer solution has $x^* = 1$, $r^* = 10$, $s^* = 0.1$ (only on one of the two copies of the arc), $r^*_{min} = 10$ and $t^* = 0.1$, giving the same value. However, the solution of the continuous relaxation using (28) in place of (19) and (22) rather has $x^* = 1/2$, $r^* = 3.75$ and $s^* = t^* = 0.25/3.75 \approx 0.066667$, yielding a smaller objective function value of 7.5. The reason behind this difference is that, at optimality, $s_{ij} = 1/r_{ij}$ does *not* hold, as (27) would require; rather, $x^2_{ij}/r_{ij} < 1/r_{ij}$. Hence, the term "$1/\min\{\, r_{ij} : (i,j) \in p \,\}$" in (5) is underestimated by the maximum of the s_{ij}.

Yet, consider instead the same numerical example except that $\bar{c} = \bar{c}_{max} = 20$, and $\delta = 4$. The optimal solution to the continuous relaxation of the original formulation now has $x^* = 1/2$, $r^* \approx 2.287135$, $s^* \approx 0.109307$ (on both arcs), $r^*_{min} \approx 12.287134$ and $t^* \approx 0.081386$, for an objective function value of ≈ 4.574270. The somewhat surprising part of this solution is that $r^*_{min} \not\leq r^*$; this is due to the form of (22), that for $x^* > 0$ allows $r^*_{min} > r^*$. Conversely, the optimal solution if (28) is used has $x^* = 1/2$, $r^* = 2.5$, $s^* = t^* = 0.1$, for a larger objective function value of 5; hence, in this case the term "$1/\min\{\, r_{ij} : (i,j) \in p \,\}$" is better approximated by the maximum of the s_{ij}, since the "big-M" $\bar{c}_{max}$ allows r^*_{min} to be (much) larger than the minimum of the r_{ij}, and therefore t to be (much) smaller than it should. Incidentally, in this case the optimal integer value is ≈ 6.66666, hence neither of the two formulations is tight.

Thus, replacing (19) and (22) by (28) makes the model slightly more compact and it is advantageous terms of bound in some cases, but it is disadvantageous in others. There is also another possibility: *add* (28) *to the formulation without eliminating* r_{min}, (19) *and* (22). This has the potential of improving the lower bounds, and it is clearly at least as tight as any of the two formulations individually; however, it is larger than both. Actually, it can be observed that when (22) is present, the relatively many ($|A|$) constraints (21) are logically redundant; in order to reduce the size of the formulation, they could be dispensed with. This is not possible if (22) is replaced by (28). All in all, we can consider 5 alternative formulations:

(a) (16)–(20), (22): the original formulation of [7] save for dispensing with (21);

(b) (16)–(22): the original formulation of [7];

(c) (16)–(20), (22), (28): the original formulation of [7] save for exchanging (21) with (28);

(d) (16)–(22), (28): the original formulation of [7] with (28) added;

(e) (16)–(21) with (28) in place of (19).

The only way to choose among the five variants is to experimentally evaluate them, which is done in Table 1. However, during the experiments formulations (c) and (d) showed, somewhat surprisingly, to be considerably less numerically stable than the other three: in particular, in a nontrivial fraction of the instances the solver declared that no feasible solution existed, whereas feasible solutions could indeed be found with the other formulations. We tried different options with scaling the constraints and setting the solver accuracy parameters,

but none were able to rectify the issue. Furthermore, the formulations (c) and (d) were not particularly efficient computationally; this is why in the table we only report results for variants (a), (b) and (e).

The Table reports the average and maximum computing time and number of nodes of the branch-and-bound tree, aggregated across a large set of instances because they were very consistent across different topologies. In particular, we could afford to aggregate all 10 `Garr` topologies and 19 `Sndlib` topologies; for each of these we performed 200 runs with different network configurations, as described in Section 6 to which we refer for further details of the instances and of the experimental set-up. We instead report more detailed results (though, again, still aggregated over 200 runs) for each of the 2 randomly-generated `Waxman` topologies because they are more significant, being much larger than the others.

Table 1: Comparing different formulations

instance	model	time		nodes	
		avg	max	avg	max
w1-100	(a)	0.85	15.98	3.53	112.60
	(b)	0.60	18.45	2.15	42.92
	(e)	0.39	6.43	2.14	34.50
w1-200	(a)	12.42	149.56	24.82	2969.85
	(b)	8.68	158.39	12.32	322.82
	(e)	3.54	108.91	12.02	161.99
Sndlib	(a)	0.01	0.09	0.14	3.17
	(b)	0.01	0.07	0.15	3.27
	(e)	0.01	0.13	0.12	4.12
Garr	(a)	0.02	0.08	0.00	1.03
	(b)	0.02	0.09	0.00	1.01
	(e)	0.02	0.12	0.02	3.61

The Table shows that on real-world topologies all formulations are roughly equivalent. However, the differences are significant for the larger `w1-100` and `w1-200` topologies. Dispensing with (21), while making the model smaller, worsens the bound quality and ultimately results in worse performance. Conversely, substituting (19) with (28) yields significant performance improvements, indicating that the formulation (28) of "$1/r_{min}$" is tighter in practice (at least, on our test bed). Therefore, in what follows, we will use formulation (e) for all the latency models we develop. In view of §5.1 below, this is actually good news, because a modeling trick akin to (28) will be necessary when dealing with guaranteed (as opposed to reserved) rates.

We now discuss how to model the different schedulers of §2, still under (7), in this framework.

3.2 Group-Based models

The extension of (18) for GB is straightforward when using the approximation (10). The only non entirely trivial choice is whether to use the lower or the upper approximation of

the delay; however, whichever the choice, the modifications to the formulæ are trivial. In our computational tests we have used the lower approximation of the delay, which leads to just replacing (24) with

$$\theta_{ij} = 3Ls_{ij} + (2L/w_{ij})x_{ij}$$

leaving (25) unchanged. Clearly, this has little impact on the shape of the optimization model, and the same would hold if using the upper approximation. Indeed, the latter choice would be the one to take if one requires certainty that the WCD never exceeds δ. However, as our experiments will show, even the less conservative choice above leads to a model that delivers significantly more costly solutions than the others. Hence we elected to keep this stance, in the belief that the computational cost of the more conservative approach should reasonably be very comparable.

As already mentioned, no worst-case delay formulæ of GB approximation have been devised yet, and it is deemed very unlikely that they could be in a future. As an advantage, no "explicit" admission control mechanism is required in this case, save for appropriately setting $\bar{c}_{ij}$. Note that, instead, one could consider employing the more accurate version (9) instead of (10). While this might indeed be possible, it would significantly complicate the mathematical model; in light of the obtained results (cf. §6) we have chosen not to pursue this approach in this work.

3.3 Weakly Rate Proportional models

We now study the latency model (12), which involves the term $|P(i,j)|$. Again, the extension of (18) is straightforward: just replace (24) with

$$\theta_{ij} = Ls_{ij} + (L/w_{ij})|P(i,j)|x_{ij} \tag{29}$$

leaving (25) unchanged. Because $|P(i,j)|$ is a constant, this again has little to no impact on the shape of the optimization model.

However, in this case *admission control* is required. This is because the delay of an existing flow q is increased on all the arcs that are used by the new flow, i.e., where $x_{ij} = 1$. Fortunately, it is easy to take this into account. One just has to define the *delay slack* of the existing flow as

$$\bar{\delta}^q = \delta^q - \frac{\sigma^q}{r^q_{min}} - \sum_{(i,j)\in q} \left(\frac{L}{r^q_{ij}} + (|P(i,j)| - 1)\frac{L}{w_{ij}} + l_{ij} + n_i \right) \ ; \tag{30}$$

this is the maximum extra delay that q can tolerate without violating the corresponding WCD constraint. The term $|P(i,j)|-1$ in (30) is due to the fact that "$|P(i,j)|$" in (12) does *not* count q, as it represents the network status *before* routing it; when defining $\bar{\delta}^q$, instead, flow q has already been routed, hence $q \in P(i,j)$ for all $(i,j) \in q$ (which in particular means that $|P(i,j)|-1 \geq 0$). Now, to ensure that the increase in q's delay—using the *fixed* reserved rates r^q_{ij} —does not exceed $\bar{\delta}^q$, it is enough to add the simple *admission control constraint*

$$\sum_{(i,j)\in q}(L/w_{ij})x_{ij} \leq \bar{\delta}^q \ . \tag{31}$$

Hence, in this case admission control can be implemented at the relatively minor cost of adding one linear constraint (in the binary variables x only) for each existing flow.

3.4 Frame-Based models

Comparing FB's latency to SRP's (8), we observe that the former involves a "simple" extra term $(L/w_{ij})|P(i,j)|$ (i.e., the one found in (12) as well), which only depends on the path (hence on variables x_{ij}), *and* a "complex" term

$$\frac{L}{w_{ij}} \frac{w_{ij} - r_{ij}}{\min\{\, r_{ij}\,,\, r_{ij}^{min}\,\}} \tag{32}$$

which is unique to (14). Clearly, (32) *only applies if* $x_{ij} = 1$, i.e., arc (i,j) is chosen to be in the path for the new flow; in fact, the term would otherwise evaluate to $+\infty$ when $x_{ij} = 0 \Longrightarrow r_{ij} = 0$.

It is easy to verify that (32) is *not* a jointly convex function in r_{ij} and all the r_{ij}^q (r_{ij}^{min}), but luckily the latter are *fixed* in this setting and therefore so is r_{ij}^{min}; this makes it convex in r_{ij}. To see that, consider the (i,j) index fixed for notational convenience and write the function

$$\phi(r) = (w - r)/\min\{\, r\,,\, r^{min}\,\}$$

that describes it (up to a constant). It is easy to prove that $\phi(r)$ is convex: just rewrite it as

$$\phi(r) = \begin{cases} \phi_1(r) = w/r - 1 & \text{if } r \le r^{min} \\ \phi_2(r) = (w - r)/r^{min} & \text{if } r \ge r^{min} \end{cases}$$

$$\text{where} \qquad \phi_1'(r) = -w/r^2 \qquad \text{and} \qquad \phi_2'(r) = -1/r^{min} \quad .$$

Clearly, both ϕ_1 and ϕ_2 are individually convex; hence, the only place in which things can go wrong is the point $r = r^{min}$ where ϕ is nondifferentiable (but clearly continuous). However

$$\phi_1'(r^{min}) = -\frac{w}{(r^{min})^2} = -\frac{1}{r^{min}}\left(\frac{w}{r^{min}}\right) \le -\frac{1}{r^{min}} = \phi_2'(r^{min})$$

because $w/r^{min} \ge 1$; that is, the left derivative in r^{min} is smaller than the right derivative, i.e., the derivative is globally non-decreasing, and therefore ϕ is convex.

This immediately allows us to construct a first MI-SOCP formulation of the problem using the classical "variable splitting" approach to represent a convex piecewise function; this ends up with the following fragment of formulation

$$\theta_{ij} = Ls_{ij} + Lv_{ij} + \frac{L}{w_{ij}}\left[\, |P(i,j)|x_{ij} - \frac{r_{ij}}{r_{ij}^{min}} \,\right] \qquad (i,j) \in A \tag{33}$$

$$r_{ij} = r'_{ij} + r''_{ij} \qquad (i,j) \in A \tag{34}$$

$$\rho x_{ij} \le r'_{ij} \le r_{ij}^{min} x_{ij} \qquad (i,j) \in A \tag{35}$$

$$0 \le r''_{ij} \le (\bar{c}_{ij} - r_{ij}^{min})x_{ij} \qquad (i,j) \in A \tag{36}$$

$$v_{ij} r'_{ij} \ge x_{ij}^2 \;\;,\;\; v_{ij} \ge 0 \qquad (i,j) \in A \tag{37}$$

where we also include the standard conic constraints (25). In particular, (34) describes the classical "splitting" of the variable r_{ij} into the sum of the variable r'_{ij} for the interval

$[\rho, r_{ij}^{min}]$ and r''_{ij} for the interval $[r_{ij}^{min}, w_{ij}]$ as dictated by (36). Then, s_{ij} and v_{ij} represent the term $1/r_{ij}$ and $1/r'_{ij}$ (if $x_{ij} = 1$, and 0 otherwise), respectively, thanks to the usual SOCP constraints.

While the above formulation is functional, it is somewhat unsettling that it contains *two* conic constraints to represent what is in fact *the same* $1/r_{ij}$ term, except that in one case its argument is bounded at r_{ij}^{min} while in the other it can go all the way up to w_{ij} (in fact, $\bar{c}_{ij}$). This can be somewhat improved upon with a simple observation: not only $\phi_1(r^{min}) = \phi_2(r^{min})$, but also $\phi_1(w) = \phi_2(w)[= 0]$. It is then immediate to see geometrically, and easy to verify algebraically, that $\phi_2(r) \geq \phi_1(r)$ for all $r \in [r_{ij}^{min}, w_{ij}]$, whereas $\phi_1(r) \geq \phi_2(r)$ for $r \in (0, r_{ij}^{min}]$. Hence, in the interval $[\rho, w]$ that matters for our problem one can alternatively define

$$\phi(r) = \max\{\, \phi_1(r)\ ,\ \phi_2(r)\,\} \tag{38}$$

thus paving the way for the reformulation of the fragment (33)–(37) using the standard representation of convex max-functions

$$\theta_{ij} = L s_{ij} + v_{ij} + \frac{L}{w_{ij}} |P(i,j)| x_{ij} \qquad (i,j) \in A \tag{39}$$

$$v_{ij} \geq L s_{ij} - L/w_{ij} \ \ ,\ \ v_{ij} \geq (L/r_{ij}^{min}) x_{ij} - L r_{ij}/(w_{ij} r_{ij}^{min}) \ \ ,\ \ v_{ij} \geq 0 \qquad (i,j) \in A \tag{40}$$

(again, including (25) as well). Constraints (40) implement (38) at the only cost of introducing one extra variable and two linear constraints; note that multiplying the constant term w_{ij}/r_{ij}^{min} of ϕ_2 by x_{ij} in (40) is necessary because it allows v_{ij} to be 0 when $r_{ij} = x_{ij} = 0$. The fragment (39)–(40) is more compact than (33)–(37), in requiring two less variables—although the preprocessor in general-purpose MI-SOCP solvers would typically use (34) to eliminate one of them—and above all one less conic constraint, for each arc; thus it appears a more promising formulation.

As can be expected, admission control constraints are significantly more complex with FB schedulers than with WRP ones, due to the presence of the nonlinear term (32). However, as already mentioned, the latency of FB is the same of WRP apart from that contribution; hence, one can at least use the same definition of delay slack (30). In fact, the extra term depends on the choices made for the new flow, and is therefore not constant (i.e., it cannot be included in the RHS of the constraint). In other words, one can write the admission control constraint as

$$\sum_{(i,j)\in q} \frac{L}{w_{ij}} \left(x_{ij} + \frac{w_{ij} - r_{ij}^q}{\min\{\, r_{ij}\ ,\ r_{ij}^{min}\,\}} \right) \leq \bar{\delta}^q \tag{41}$$

with the $\bar{\delta}^q$ of (30). This is clearly related to the WRP version (31), but for the extra part (32); it has to be remarked, again, that that term only applies when $x_{ij} = 1 \Longrightarrow r_{ij} > 0$. Exploiting the already discussed properties of (32), a SOCP formulation can be easily obtained:

$$\textstyle\sum_{(i,j)\in q} (L/w_{ij}) \big(x_{ij} + (w_{ij} - r_{ij}^q) z_{ij} \big) \leq \bar{\delta}^q \tag{42}$$

$$z_{ij} \geq 1/r_{ij}^{min} \ \ ,\ \ z_{ij} \geq s_{ij} \qquad (i,j) \in q \tag{43}$$

Note that the "$1/r_{ij}^{min}$" term cannot cause any problems here: because $(i,j) \in q$, $(i,j) \notin A'$ and therefore $r_{ij}^{min} > 0$. It is also important to remark that *neither the s_{ij} nor the z_{ij} depend on the flow q*; that is, these can be defined just once for all arcs $(i,j) \in A$ and then used to define the admission constraints for all the flows. Actually, the z_{ij} variables only need to be defined for all $(i,j) \in A''$, i.e., for which at least one existing flow is routed.

4 Modeling guaranteed rates in the link latency

We now discuss how to modify the previously proposed models to remove the bound assumption (7). This is nontrivial, since guaranteed rates have to substitute reserved rates in two different places: the expression of the link latency θ_{ij}, and the formula for the overall end-to-end delay. These two aspects can be partly separated; in fact, one can, in principle, write models where only one of the two improvements is performed. In this Section we will only concentrate on the impact on the link latency expressions, while the impact on the overall end-to-end delay expression will be explored in the next one.

4.1 Strictly Rate Proportional models

To extend the SRP model to using the more accurate formula (6), one just has to use (3) to get

$$\theta_{ij} = \frac{L}{w_{ij}} + \begin{cases} \dfrac{L}{w_{ij}}\left(\dfrac{\bar{r}_{ij}}{r_{ij}} + 1\right) & \text{if } (i,j) \in A'' \\ 0 & \text{otherwise} \end{cases} \tag{44}$$

Thus, employing (44) instead of (8) in the models is relatively straightforward; some arcs (the "empty" ones) actually have constant latency (in fact, $\bar{r}_{ij} = 0 \Longrightarrow g_{ij} = w_{ij}$), for the others the latency (besides a yet different constant) has the same "$1/r_{ij}$" form, with just a different scaling factor. Hence, the optimization problem is largely unaffected: one only has to replace (26) with

$$t + \sum_{(i,j)\in A} \bar{l}_{ij} x_{ij} + \sum_{(i,j)\in A''} \left(\frac{L\,\bar{r}_{ij}}{w_{ij}} s_{ij} + \frac{L}{w_{ij}} x_{ij}\right) \leq \delta \tag{45}$$

which is in no way significantly more complex. Actually it is somewhat less so, since *for arcs in A' the variable s_{ij} and the corresponding conic constraint* (25) *are not needed*.

However, removing assumption (7) also implies that the delay depends on the other flows (via the term $\bar{r}_{ij}$ in (44)), and therefore requires admission control. Indeed, consider an existing flow q, some fixed $(i,j) \in q$, and assume that the new flow is passing through (i,j) (which implies $x_{ij} = 1$) with some given reserved rate r_{ij}; note that this means that, "from the viewpoint of q", $P(i,j)$ is nonempty because it contains at least the new flow. The arrival of the new flow does impact the delay formula (44). Indeed, let $\bar{r}_{ij}^{-q} = \sum_{h\in P(i,j)\setminus\{q\}} r_{ij}^h$ be the sum of all the rates of existing flows on (i,j) save for q: (44) reads

$$\theta_{ij}^q = \frac{L}{w_{ij}}\left(\frac{\bar{r}_{ij}^{-q} + r_{ij}}{r_{ij}^q} + 2\right) \quad ,$$

since "from the viewpoint of q", the sum of the "other" rates (which enters the definition of the guaranteed rate for q) is $\bar{r}_{ij}^{-q} + r_{ij}$. This means that the increase in latency due to the new flow passing through (i,j) is

$$\Delta\theta_{ij}^{q} = \begin{cases} \dfrac{L\, r_{ij}}{w_{ij}\, r_{ij}^{q}} & \text{if } |P(i,j) \setminus \{q\}| > 0 \\ \dfrac{L}{w_{ij}} \left(\dfrac{r_{ij}}{r_{ij}^{q}} + 1 \right) & \text{otherwise} \end{cases} .$$

In other words, let q be partitioned into $q' \cup q''$, where q' contains the arcs (i,j) that are "empty but for q" ($P(i,j) = \{q\}$) and q'' those that also contain other flows. We can define the delay slack $\bar{\delta}^q$ (≥ 0) of flow q as

$$\bar{\delta}^q = \delta^q - \frac{\sigma^q}{r_{min}^q} - \sum_{(i,j)\in q} \bar{l}_{ij} - \sum_{(i,j)\in q''} \frac{L}{w_{ij}} \left(\frac{\bar{r}_{ij}^{-q}}{r_{ij}^{q}} + 1 \right) . \tag{46}$$

One can then ensure that the delay of flow q remains below its maximum threshold by, again, simply adding the linear admission control constraint

$$\sum_{(i,j)\in q} \frac{L}{w_{ij} r_{ij}^{q}} r_{ij} + \sum_{(i,j)\in q'} \frac{L}{w_{ij}} x_{ij} \leq \bar{\delta}^q \tag{47}$$

to the model. Thus, removing assumption (7) for SRP schedulers hardly changes the model, save for the introduction of as many linear constraints as there currently are flows in the network.

4.2 Weakly Rate Proportional models

Also for the WRP latency model (11), removing assumption (7) leads to a model hardly more complex: from (3) we have

$$\theta_{ij} = |P(i,j)| \frac{L}{w_{ij}} + \frac{L}{w_{ij}} \left(\frac{\bar{r}_{ij}}{r_{ij}} + 1 \right) \tag{48}$$

which easily gives

$$t + \sum_{(i,j)\in A} \left[\frac{L\, \bar{r}_{ij}}{w_{ij}} s_{ij} + \left(|P(i,j)| \frac{L}{w_{ij}} + \bar{l}_{ij} \right) x_{ij} \right] \leq \delta \tag{49}$$

as the correct substitute for (18). Note that in (49) the coefficient $\bar{r}_{ij}$ can be zero (it is in particular so for arcs in A'), and therefore once again some of the auxiliary variables s_{ij}, and the corresponding conic constraints, can be avoided.

As far as admission control is concerned, things are, again, only slightly more complex than in the bound case. In fact, in this case the increase in latency for an existing flow q due to the passing of the new flow from $(i,j) \in q$ is

$$\Delta\theta_{ij}^{q} = \frac{L}{w_{ij}} \left(\frac{r_{ij}}{r_{ij}^{q}} + 1 \right) . \tag{50}$$

It is then only necessary to define the delay slack as

$$\bar{\delta}^q = \delta^q - \frac{\sigma^q}{r^q_{min}} - \sum_{(i,j)\in q} \left[\frac{L}{w_{ij}} \left(\frac{\bar{r}^{-q}_{ij}}{r^q_{ij}} + |P(i,j)| \right) + l_{ij} + n_i \right] \ , \tag{51}$$

where note that $P(i,j)$ here contains q, while in (48) $P(i,j)$ is intended as the set of existing flows in (i,j) save q, which justifies for the apparently missing "+1" term. Then, the admission control constraint is once again linear:

$$\sum_{(i,j)\in q} \frac{L}{w_{ij}} \left(\frac{r_{ij}}{r^q_{ij}} + x_{ij} \right) \leq \bar{\delta}^q \ . \tag{52}$$

4.3 Frame-Based models

Removing assumption (7) for FB schedulers leads to formulæ similar to the bound case, although with some differences. Indeed, from (13) the FB formula is

$$\theta_{ij} = \frac{L}{w_{ij}} \left[\frac{\bar{r}_{ij}}{\min\{ r_{ij} , r^{min}_{ij} \}} + \frac{\bar{r}_{ij}}{r_{ij}} + |P(i,j)| + 1 \right] \ , \tag{53}$$

i.e., the same as (48) plus the extra term

$$\frac{L}{w_{ij}} \frac{\bar{r}_{ij}}{\min\{ r_{ij} , r^{min}_{ij} \}} \ . \tag{54}$$

It is important to remark that (54) only concern arcs in A'': in fact, for $(i,j) \in A'$ one has $\bar{r}_{ij} = 0$. This is very convenient because for $(i,j) \in A'$ one would also have $r^{min}_{ij} = 0$, leading to the risk that (54) be ill-defined (when $r_{ij} = 0$, which however implies $x_{ij} = 0$). Fortunately, this is, actually, somewhat "less complex" than (32). Indeed, disregard again the (i,j) index; now, the relevant function is

$$\phi(r) = 1/\min\{ r , r^{min} \}$$

that is again clearly convex as

$$\phi(r) = \begin{cases} \phi_1(r) = 1/r & \text{if } r \leq r^{min} \\ \phi_2(r) = 1/r^{min} & \text{if } r \geq r^{min} \end{cases}$$

$$\text{with} \qquad \phi'_1(r) = -1/r^2 \qquad \text{and} \qquad \phi'_2(r) = 0 \ ,$$

so that it's immediate to realize that $\phi'_1(r^{min}) < \phi'_2(r^{min})$. We can then quickly come up with a SOCP formulation, directly in the "max" format, because it is immediately clear that, again, (38) holds. Hence, one can represent (54) simply by means of a variable v_{ij} subject to the obvious linear constraints

$$v_{ij} \geq s_{ij} \ , \ v_{ij} \geq 1/r^{min}_{ij} \ , \ (i,j) \in A'' \ .$$

Note once again that these are only defined for "nonempty" arcs, which is crucial since $1/r_{ij}^{min}$ is ill-defined for $(i,j) \in A'$. This having been said, (18) becomes

$$t + \sum_{(i,j)\in A} \left(|P(i,j)| \frac{L}{w_{ij}} + \bar{l}_{ij} \right) x_{ij} + \sum_{(i,j)\in A''} \frac{L\,\bar{r}_{ij}}{w_{ij}} (s_{ij} + v_{ij}) \leq \delta \quad .$$

Note that the formula is actually the same as (49) (obviously, due to the similarity of the two delay formulaæ) except for the term in v_{ij}; indeed, as already remarked even (49) could have been written with the term in s_{ij} only appearing for arcs in A'' ($\bar{r}_{ij}$ being null in A'). In particular, this means that also in this case one does not need to define the variables s_{ij} for $(i,j) \in A'$, and analogously for the v_{ij}.

Regarding admission control, the latency increase due to the new flow can be seen as composed by two distinct parts: the one corresponding to the "WRP part" of the latency formula, and the one corresponding to (54). The first leads to an increment of the delay with exactly the same form of (50). For the second, however, the increment is not linear. Thus, we can again define the delay slack $\bar{\delta}^q$ by (51) exactly as in the WRP case, since the extra term depends on the rates of the newly added flow and therefore, not being fixed, it cannot be counted in the RHS of the constraint. All this leads to

$$\sum_{(i,j)\in q} \frac{L}{w_{ij}} \left(\frac{\bar{r}_{ij}^{-q} + r_{ij}}{\min\{\, r_{ij}\,,\, r_{ij}^{min} \,\}} + \frac{r_{ij}}{r_{ij}^{q}} + x_{ij} \right) \leq \bar{\delta}^q \quad . \tag{55}$$

Note that, at the denominator, r_{ij}^q is "counted" in r_{ij}^{min}; anyway, the denominator is just the minimum of r_{ij}^h for *all* h (including q and the "new" flow). Now, the analysis follows well-established steps: writing

$$\phi(r) = \begin{cases} \phi_1(r) = (\bar{r} + r)/r & \text{if } r \leq r^{min} \\ \phi_2(r) = (\bar{r} + r)/r^{min} & \text{if } r \geq r^{min} \end{cases}$$

one rapidly realizes that $r \leq r^{min} \Longrightarrow \phi_1 \geq \phi_2$, and therefore that (38) holds. This means that (55) can be expressed as the SOCP fragment

$$\sum_{(i,j)\in q} \frac{L}{w_{ij}} \left(z_{ij} + \frac{r_{ij}}{r_{ij}^{q}} + x_{ij} \right) \leq \bar{\delta}^q \tag{56}$$

$$z_{ij} \geq (\bar{r}_{ij} + r_{ij})/r_{ij}^{min} \quad , \quad z_{ij} \geq \bar{r}_{ij} s_{ij} + 1 \qquad\qquad (i,j) \in q \tag{57}$$

Again, (56) is (52) plus the extra term necessary to deal with (54), which requires the new variable z_{ij}; again, these are "shared" among all admission control constraints, and need only be defined for arcs in A''. This is in particular relevant for two reasons. On one hand, for $(i,j) \in A'$ one has $r_{ij}^{min} = 0$, which leaves the first constraint in (57) ill-defined. On the other hand, the second part of (57) is due to represent $z_{ij} \geq \phi_1(r_{ij}) = \bar{r}_{ij}/r_{ij} + 1$. However, one has to remind that the new flow impacts the delay on arc (i,j) only if it actually traverses it, i.e., $x_{ij} = 1$; if not, one has to take $\phi_1 \equiv 0$, i.e., $\phi = \phi_2$. Yet, $z_{ij} \geq 1$ even if $x_{ij} = 0$. Still, this is actually not a problem since $(i,j) \in q \Longrightarrow (i,j) \in A''$, hence $\bar{r}_{ij} \geq r_{ij}^{min}$ and therefore $z_{ij} \geq 1$ anyway.

5 Modeling guaranteed rates in the end-to-end delay

In the previous section we have assumed guaranteed rates to only affect the link latency expression. However, as per (4) and (5), this is not so: guaranteed rates are also involved in both the lower bound and minimum rate expressions.

5.1 Extending the basic formulation

Writing (4) in terms of the guaranteed rate is easy: plugging in (3) yields

$$(\rho\ \bar{r}_{ij})x_{ij} \leq (w_{ij} - \rho)r_{ij} \quad , \tag{58}$$

that can directly replace (21): the left-hand side again is 0 if $x_{ij} = 0$, which does not disrupt the semi-continuous status of r_{ij}. The right-hand side of (4) does not need to be changed, as it is in fact related to the reserved rate, rather than to the guaranteed one.

As far as (5) is concerned, only the leftmost term is affected: the others are either constant or depend on the scheduling algorithm, and therefore have already been treated in §4. Here, one can readily extend the approach of (27)–(28) in §3.1:

$$\frac{1}{\min\{\,(w_{ij}r_{ij})/(\bar{r}_{ij} + r_{ij})\ :\ (i,j) \in p\,\}} = \max\left\{\,\frac{\bar{r}_{ij} + r_{ij}}{w_{ij}r_{ij}} = \frac{\bar{r}_{ij}}{w_{ij}r_{ij}} + \frac{1}{w_{ij}}\ :\ (i,j) \in p\,\right\} \quad .$$

In other words, we exploit the fact that the *inverse of the guaranteed rate is linear in the inverse of the reserved rate*, together with the fact that *we always have the inverse of the reserved rate (approximately) represented by* s_{ij}. Hence, we can play the same trick as in (28), eliminating r_{min} (actually, g_{min}) and the corresponding constraints, and adding

$$t \geq \frac{\bar{r}_{ij}}{w_{ij}} s_{ij} + \frac{1}{w_{ij}} x_{ij} \qquad (i,j) \in A \tag{59}$$

to replace them. This can be done uniformly in all formulations of §4, irrespectively of the constraints on θ_{ij}, i.e., on the choice of the scheduler. In view of the computational results in §3.1, this formulation can be expected to be effective in practice. We suspect that it might be possible to construct a MI-SOCP formulation with an explicit g_{min} variable, but following the guidelines already developed for the bound versions, in our computational results we have consistently used (59) to represent the "σ/g_{min}" term in the end-to-end delay constraint.

Actually, (59) could be used as well in all the "bound" formulations of §3; i.e., one could, in principle, represent the contribution to the WCD of the "σ/g_{min}" term in (5) using guaranteed rates, while using reserved rates to estimate the link latencies. However, we did not find this modeling variant to have any convincing rationale, hence it has not been computationally tested.

5.2 Extending the admission control constraints

Using guaranteed rates in the end-to-end delay expression affects the admission control constraints. Fortunately, the impact is almost independent of the specific scheduler used.

The observation is that all admission control constraints rely on the definition of the *delay slack*, which is slightly different according to the case: cf. (30), (46), and (51). However, all three forms contain the term "σ^q/r^q_{min}": since this now has to be σ^q/g^q_{min}, and g^q_{min} depends on all the other flows (comprised the "new" one currently being routed), that term is no longer a constant. This is not hard to solve, though: one can simply remove the term "σ^q/r^q_{min}" from the corresponding expression of $\bar{\delta}^q$, and substitute it with $\sigma^q t^q$, where:

$$t^q \geq \frac{\bar{r}_{ij}}{w_{ij} r^q_{ij}} + \frac{r_{ij}}{w_{ij} r^q_{ij}} \qquad (i,j) \in q \ . \tag{60}$$

This exploits the fact that *the inverse of the guaranteed rate of a flow q is linear in the reserved rate of every other flow, comprised the one just to be routed*, if its own rate is fixed. Hence, admission control constraints now require one extra variable for each flow, and as many linear constraints as there are arcs in the corresponding path. This increases the size of the models, but it does not dramatically change their shape. Also, note that the first term in the right-hand-side of (60) is actually constant; thus, that part can be left in the definition of the delay slack $\bar{\delta}^q$, leaving t^q to be constrained only by the term linear in r_{ij}.

All in all, we have shown how a trade-off potentially exists between more accurate models of the different scheduling algorithms and the size and complexity of the corresponding optimization models. This trade-off is computationally explored in the next section.

6 Computational results

We now report our computational experience aimed at comparing the performance of the different MI-SOCP formulations. All the experiments have been performed on a 2.299 Ghz AMD Opteron 6376 with 16Gb RAM, running a 64 bits Linux operating system (Ubuntu 12.4). The MI-SOCP models were solved by the state-of-the-art, off-the-shelf, commercial solver `Cplex 12.6`, called via the C API. Basically all the running time was spent in the solver.

6.1 Generating the instances

To generate the instances we followed the same process of our earlier paper [7], which we summarize here. We considered three sets of IP network topologies. Two of them are real-world networks: the `GARR` subset [1] of the Internet Topology Zoo [2] and the `SNDlib` topologies [3], which can be downloaded in `gml` format. Since these networks all have less than 100 nodes, we also produced two random larger topologies generated according to the realistic *Waxman* model [31]. These are produced directly by the `FNSS` tool [26], which can both read `gml` topology files and generate random ones according to different models, simply by specifying the number of nodes n and the (probability) parameter $\alpha \in (0, 1]$ representing the *link density*: for our `Waxman` topologies we choose $n \in \{100, 200\}$ and $\alpha = 0.4$.

`FNSS` was also used to assign realistic link capacities and generate realistic traffic matrices that take into account the selected link capacities. These have been chosen among $\{1, 10, 40\}$ Gbps, according to the "edge betweeness" algorithm. For the traffic matrices we had to specify the mean traffic demand $\mu(T)$ and its standard deviation $\sigma^2(T)$; for our experiments

we selected $\mu(T) = 0.8$ Gbps and $\sigma^2(T) = 0.05$. The MTU L was set to 1500 bytes, since nearly all IP over Ethernet implementations use the Ethernet V2 frame format. Node delays n_i and link delays l_{ij} were set equal to L/w_{ij}. The reservation capacities c_{ij} were taken to be all equal to the corresponding link capacity w_{ij}. Flow bursts σ were set to 3 times the MTU value. Finally, to define flow deadlines δ, we computed—using the *bound* latency formulæ of SRP—the least possible value δ_{min}, under which no routing is possible, and the maximum possible value δ_{max}, over which the delay constraint becomes redundant. Then, δ was randomly chosen uniformly within the interval $[\,\delta_{min}\,,\,\delta_{min} + (\delta_{max} - \delta_{min})\beta\,]$ for a fixed parameter β; for our experiments we used $\beta = 0.2$, which should produce "tightly constrained" flows.

All the above parameters were used to perform network simulations, which in turn produced different instances of the ADCSP problems corresponding to different sets of active flows, that we will call *statuses*. The details of the simulations are described in [9], where a first attempt at evaluating the trade-off between the cost of a scheduler and the corresponding network performance (typically measured by the blocking probability) is performed; we refer the interested reader there for details. In a nutshell, however, in the simulations we assume that each flow in the generated traffic matrix randomly performs admission requests, and (if admitted) lasts for some random period of time. Flow request inter-arrivals are exponential with varying rate λ (the "load" of the network), and each flow lasts for an exponentially distributed time with mean equal to 1 sec. The four different load values 0.1, 1, 10 and 100 were used in order to test the latency models at different levels of network congestion: the higher the load, the more active flows (hence, admission control constraints) are typically present. The simulations have been performed assuming FB schedulers with the "bound" version (13) of the latency formula, and using reserved rates to define the WCD constraint, i.e., under the bound assumption (7). The rationale for this choice is that this particular latency model is the most "conservative" one in terms of admission control. This enables us to ensure that the corresponding network configuration satisfies all admission control for the other latency models, thereby avoiding to produce a very large number of unfeasible instances (flow requests that cannot be accepted).

Then, a network status is simply the subset of the flows of the traffic matrix that is occupying the network at some given point of the simulation, each of them characterized by a path and the rate allocations on the corresponding links. We divided the simulation time into 20 time slots, of which we only used the last 10 ones to make sure that transitory effects are over; then, the status is just the set of active flows at the beginning of one of these time slots. For each topology and for each of the 4 load values we repeated our simulations with 5 different seeds; hence, we have 50 instances for each load value, for a grand total of 200 instances for each topology.

6.2 Computational results

The first set of results we report concern the impact on the computational cost and the usage of network resources (i.e., the reserved rates) of the different representations of delay, i.e., employing or not the bound assumption (7). We tested three possible modeling variants:

- the *bound* (B) models of §3, in which the bound assumption (7) is used in both the

latency and the WCD constraint;

- the *semi-worst-case* (S) models, obtained replacing the bound latency formulæ with the worst-case ones, as described in §4, but leaving the WCD constraint defined with reserved rates;
- the *worst-case* (W) models, obtained by using guaranteed rates in both the latency formulæ and the WCD constraint, as advocated in §5.

As previously discussed, a fourth variant would be possible but we elected not to test it. The results are reported in Tables 2, 3 and 4. Each row corresponds to the average results of 600 instances, corresponding to 50 samples, 4 loads, and the 3 scheduler classes SRP, WRP and FB. In the tables, columns "time" and "nodes" report the average (of the average and maximum, respectively) solution time and branch-and-bound nodes required to solve each instance. Column "failed" reports the average ratio between the number of "failed" flows, i.e., those that were not admitted (the corresponding ADCSP instance was unfeasible) for the given model and the maximum number of failed flows amongst all other models. Similarly, column "rate" is the average of the ratio between the amount of allocated rate, for a given instance, for that model and the maximum allocated rate, for that same instance, amongst all other models. Both figures provide some indication about how efficient, in terms of network resources usage, a given model is. Indeed, a more accurate representation of the delay allows the same WCD guarantee to be met with a lower rate allocation, which therefore leaves more headway for subsequent flows to be routed. Similarly, a lower failure rate means that, in a given status, a more accurate representation of the delay allows a feasible routing to be found, whereas this is not possible if less accurate representations are used. Note that we can compute the allocated rate only for non-failed flows; hence, those that fail are assigned a relative rate of 1, corresponding to the maximum rate among non-failed flows. We found that not doing so would unduly skew the results towards schedulers that fail more often.

We immediately remark that aggregating across different scheduler classes is not an obvious choice: in principle, they can be expected to—and they do in practice—attain different results in terms of allocated rates, and therefore failures (and time, as we shall see). However, the results for the different latency classes were similar enough that aggregating them was still possible; we found that this still provided an accurate enough picture of the relative behavior of the different models, while making the tables much smaller, and therefore much easier to read (considering the large number of instances they cover). Furthermore, more detailed results are provided later on to discuss the finer nuances of the different schedulers' behavior. For this reason we did not consider GB schedulers at this stage: they are defined only for one of the model classes (B), and since their results are (as we shall see) rather worse than the others they would have considerably skewed the aggregated results.

The Tables already show that all instances corresponding to real-life topologies can be solved in split seconds with all the three models. As one would expect, failures and rates are higher for the B models, these being the most conservative ones. Interestingly, and somewhat unexpectedly, the running times of the B models are also visibly higher (although still fairly small) than those of the other classes. Despite the fact that S and W models have more constraints and variables, they are easier to solve in practice. This may be due to the fact that they are, in a different sense, *less* constrained: it is easier to find solutions for

them than for B. Indeed, the failure ratio for both S and W models is often visibly smaller than that of the B model, meaning that using the more accurate representation of the delay has a significant impact on the quality of the obtained solutions. In general, failure rates are very similar between the S and W models; however, in a few cases W models attain higher failure counts. This is due to a very limited number of cases in which W models fail to find a solution due to numerical issues. Even in the cases where the difference is visible (di-yuan and pdh), the actual number of flows that fail due to these problems is small; the large difference in ratio is due to the fact that the overall number of failures is small as well. In fact, conversely S always allocated visibly higher rates than W, as the theory dictates. Hence, computing the routings using W models should, in real-world usage, translate into (significantly) fewer failures than these of S models (and, a fortiori, of the B ones), i.e., in better network utilization, due to the fact that lower allocated rates should allow more other flows to be accepted. Validating such a claim will require extensive simulations, that are outside the scope of the present work; an initial set of results have already been obtained [9], but more study is necessary. From the running time viewpoint, W occasionally has larger cost—but only slightly so—than S, which could be expected as discussed in §5.

All these observations on smaller, real-life topologies are confirmed in the much larger, randomly-generated `Waxman` ones. Here the difference in running times is very remarkable, unexpectedly so: the mean running time of B models is reduced by over two order of magnitude by using S or W ones instead, and the maximum is reduced by well over one order of magnitude. W models are on average twice as costly as S ones, and their maximum running time is well over one order of magnitude larger than that of S models. However, the average running time is, unlike what was found in [7], compatible with real-time usage, especially considering that, except for the model tuning of §3.1, no attempt has been done to reduce running times: the solver was ran with all default parameters, and single-threaded.

A more detailed recount of the results is given in Table 5 for three specific instances (taken each from one of the three sets). In the Table, the same statistics as above are reported, but this time not aggregating the results of the three (or four) different schedulers for the same class of models.

The Table shows that GB (of which only the B version exists) has the highest failure and reserved rates; this is clearly due to the extra factor of three in the rate-dependent term. Note that we have chosen a *lower bound* approximation on purpose, so as to discount the hypothesis that GB's poor performance were due to a pessimistic upper-bound approximation. Despite this, GB always needs to reserve the largest amounts on the arcs to obtain the required level of WCD, and therefore has by far the worst failure ratio. This is also the reason why it is not the one with the largest running time: usually, unfeasible instances are solved faster than feasible ones. FB is the typically next least-efficient scheduler after GB, both in terms of failures and of reserved rates. This is also expected, due to the fact that, as discussed at length in the previous sections, its latency formula has one extra term w.r.t. the one of WRP, which in turn has one term that is usually (always, except in B models) larger than that of SRP. In other words, all other things being equal FB requires larger rates to achieve the same WCD. However, there are rare exceptions where B-FB allocates less rates than B-SRP, as shown in the `Garr` instance. This can be explained as follows: the FB and WRP formulæ contain the term "$|P(i,j)|L/w_{ij}$" where the SRP one has "L/w_{ij}". The

FB/WRP term is therefore typically larger, except in the case where there are no other flows passing through (i, j) (i.e., $(i, j) \in A'$), in which case it is smaller. Hence, there are cases in which this may lead SRP to allocate higher rates than FB. Conversely, WRP is always not worse than FB, and it can be better than SRP for the same reason; indeed, this phenomenon shows up both in the `w1-200` and the `atlanta` instance. As already discussed in §2, this does not reflect the actual behavior of the scheduler, but rather is an artifact due to the bound assumption (7). In fact, this only happens for bound formulæ; for S and W models the artifact disappears, and the natural ordering is always observed where SRP allocates the least, followed by WRP and finally FB. In our experiments, the allocations of the different schedulers for S and W models were almost always identical, with minor differences between FB and the other two, as the Table shows; the dominating factor in determining allocations is the choice between the B, S and W model, with the choice of the scheduler playing a very minor role. The same happens for running times: apart from the B model, the expected behavior consistently shows up where SRP is the fastest, followed by WRP and then by FB. However, this is visible only in the `w1-200` topology (for the W model), due to it having longer running times; in all cases, the difference is negligible.

All in all, these results indicate that modeling other classes of GPS-derived schedulers than SRP ones, while complicating the models, does not make the ADCSP problem significantly more difficult to solve, at least on real-world sized instances. Neither does using more accurate models of the latency and delay formulæ, i.e., modeling the difference between *reserved* rates and *guaranteed* ones; surprisingly, this most often results in both better performance in terms of failures and allocated rates, *and* in (sometimes considerably) shorter running times. Hence, the modeling power allowed by MI-SOCP formulations, combined with the effectiveness of state-of-the-art solvers, allows to do away with the bound assumption (7) that has invariably been employed so far. We believe this paves the way to interesting developments in this matter, as discussed in the next section.

7 Conclusions and future research

In this paper we have considerably extended the set of previously available models for the delay-constrained routing problem. The extension is threefold:

1. we have shown how to account for the latency formulæ corresponding to the most relevant classes of GPS-derived schedulers from the literature;

2. we have explicitly introduced the concept of admission control constraints, and shown how to implement them for all classes of schedulers;

3. we have shown how to model the difference between *reserved* and *guaranteed* rates, in both the latency and the delay formulæ, taking into account admission control.

In all these cases, the model remains a MI-SOCP; provided that the right modeling choices are made, this allows one to solve the problem for instances of realistic size, and also for larger randomly generated ones, in time compatible with the constraints of a real operating environment.

Our results can therefore be of interest for the actual on-line management of a communication network. First and foremost, they seem to indicate that indeed a nontrivial trade-off exists between using lower-complexity schedulers, such as GB, and the network performance. Characterizing this trade-off will, however, require actual network simulations. We have not included them in this paper because the focus here was on the relative performance of the different models in solving the *same* instances of ADCSP, which is not something that would often (if ever) happen during a simulation. Some initial results in this direction have already been obtained [9], but only limited to a subset of the models. In particular, our results here seem to indicate that the choice of employing or not the bound assumption (7) (i.e., between models B, S and W, where this is allowed—which is not for GB) may be far more relevant than the choice of the scheduler; however, this will require a nontrivial work to be experimentally confirmed.

Having established that the basic ADCSP problem is efficiently solvable for realistic instances also allows to start investigating more complex issues. In particular, all models so far—comprised the ones developed here—assume that a flow can only be admitted if doing so does not disrupt the existing ones, in particular by making them to violate their WCD constraints *with their current choice of the path and reserved rates*. It might be conceivable, however, that alternative approaches exist where one path may still be admitted provided that a limited set of changes is allowed on the existing ones. Alternatively, it may be interesting to explore scenarios in which—say, from time to time—a global re-routing phase is enacted where a *multi-flow* problem is solved in order to find solutions not affected by the sequence of myopic choices made during the successive routing of individual flows. Doing this might require to define some more sophisticated notions of the "fitness" of a global solution, in terms of its capacity to support a set of yet-unknown future flow requests, than what has been used so far (i.e., just the sum of all the reserved rates across all the arcs and flows). The need of such a notion seems in fact to emerge from the initial results of [9]. Any of these possible improvements is very likely to result in much more challenging models, due to either a much larger size, or to the fact that the proposed formulations being MI-SOCP often hinged on the circumstance that all but few of the decisions (paths, rates) were fixed, or both. Hence, it is likely that such versions of the problem would not be solvable by general-purpose tools, and would require specific algorithmic developments. All this makes, in our opinion, our results relevant and worthy of further study.

Acknowledgements

The first and second authors gratefully acknowledge the contribution of the project of the University of Pisa "Mathematical models and computational methods for complex networks". The second author has been funded by the FIRB 2013 project RBFR13ZYQL of the Italian Science and Education Ministry. This research has also been partly undertaken under the auspices of the PRIN 2012 Project "Mixed-Integer Nonlinear Optimization: Approaches and Applications" of the Italian Science and Education Ministry.

References

[1] Garr.

[2] The internet topology zoo.

[3] Sndlib.

[4] J.C.R. Bennett and H. Zhang. WF2Q: Worst-case fair weighted fair queueing. In *INFOCOM*, pages 120–128, 1996.

[5] F. Checconi, L. Rizzo, and P. Valente. QFQ: Efficient packet scheduling with tight guarantees. *IEEE/ACM Trans. Netw.*, 21(3):802–816, 2013.

[6] A. Diwan, J. Kuri, and S. Sanyal. Optimal allocation of rates in guaranteed service networks. *Informatica*, 4:201–212, 2012.

[7] A. Frangioni, L. Galli, and M.G. Scutellà. Delay-constrained shortest paths: Approximation algorithms and second-order cone models. *Journal of Optimization Theory and Applications*, 164(3), 2015.

[8] A. Frangioni, L. Galli, and G. Stea. Optimal joint path computation and rate allocation for real-time traffic. *The Computer Journal*, Vol. 58 No. 6, 2015.

[9] A. Frangioni, L. Galli, and G. Stea. Qos routing with worst-case delay constraints: Models, algorithms and performance analysis. Technical Report, Dipartimento di Informatica, Università di Pisa, 2015.

[10] A. Frangioni and C. Gentile. Perspective Cuts for a Class of Convex 0–1 Mixed Integer Programs. *Mathematical Programming*, 106(2):225–236, 2006.

[11] A. Frangioni, C. Gentile, E. Grande, and A. Pacifici. Projected Perspective Reformulations with Applications in Design Problems. *Operations Research*, 59(5):1225–1232, 2010.

[12] S.J. Golestani. A Self-Clocked Fair Queueing Scheme for Broadband Applications. *Proc. of IEEE INFOCOM'94, Toronto, Canada*, pages 636–646, 1994.

[13] A. Juttner, B. Szviatovszki, I. Mecs, and Z. Rajko. Lagrange relaxation based method for the QoS routing problem. In *Proceedings of INFOCOM'01, 22-26 April 2001, Anchorage, Alaska USA*, pages 859–868. IEEE, 2001.

[14] M. Karsten. Approximation of generalized processor sharing with interleaved stratified timer wheels. *IEEE/ACM Trans. Netw.*, 18(3):708–721, 2010.

[15] T. Korkmaz and M. Krunz. Multi-constrained optimal path selection. In *Proceedings of INFOCOM'01*, 2001.

[16] F. Kuipers, T. Korkmaz, and M. Kruntz. An overview of constraint-based path selection algorithms for QoS routing. *IEEE Communications Magazine*, 2002.

[17] L. Lenzini, E. Mingozzi, and G. Stea. Eligibility-Based Round Robin for Fair and Efficient Packet Scheduling in Interconnection Networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(3):254–266, 2004.

[18] L. Lenzini, E. Mingozzi, and G. Stea. Tradeoffs between low complexity, low latency and fairness with deficit round robin schedulers. *IEEE/ACM Transactions on Networking*, 12(4):681–693, August 2004.

[19] L. Lenzini, E. Mingozzi, and G. Stea. Performance Analysis of Modified Deficit Round Robin Schedulers. *IOS Journal of High-Speed Networks*, 16(4):399–422, 2007.

[20] D.H. Lorenz and A. Orda. Optimal partition of QoS requirements on unicast paths and multicast trees. *IEEE/ACM Transactions on Networking*, 10:102–114, February 2002.

[21] A. Lori, G. Stea, and G. Vaglini. Towards Resource-Optimal Routing Plans for Real-Time Traffic. In T. Margaria and B. Steffen, editors, *Leveraging Applications of Formal Methods, Verification, and Validation*, volume 6415 of *Lecture Notes in Computer Science*, pages 214–227. 2010.

[22] Q. Ma and P. Steenkiste. Quality-of-Service Routing for Traffic with Performance Guarantees. In *In Proc. IFIP International Workshop on Quality of Service*, pages 115–126, 1997.

[23] A. Orda. Routing with End-to-End QoS Guarantees in Broadband Networks. *IEEE/ACM Trans. on Networking*, 7(3):365–374, 1999.

[24] K. Parekh and R.G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single Node Case. *IEEE/ACM Trans. on Networking*, 1:344–357, 1993.

[25] M. Saad, A. Leon-Garcia, and W. Yu. Optimal Network Rate Allocation under End-to-End Quality-of-Service Requirements. *IEEE Transactions on Network and Service Management*, 4(3):40–49, 2007.

[26] L. Saino, C. Cocora, and G. Pavlou. A toolchain for simplifying network simulation setup. In *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*, SIMUTOOLS '13, ICST, Brussels, Belgium, Belgium, 2013. ICST.

[27] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. *IEEE/ACM Transactions on Networking*, 4:375–385, June 1996.

[28] P. Valente. Exact gps simulation with logarithmic complexity, and its application to an optimally fair scheduler. In *SIGCOMM*, pages 269–280, 2004.

[29] P. Valente. Private communication, 2014.

[30] Z. Wang and J. Crowcroft. Qos routing for supporting resource reservation. *IEEE Journal on Selected Areas in Communications*, September 1996.

[31] B. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.

[32] J. Xu and R.J. Lipton. On fundamental tradeoffs between delay bounds and computational complexity in packet scheduling algorithms. In *SIGCOMM*, pages 279–292, 2002.

[33] X. Yuan. Heuristic Algorithms for Multiconstrained QoS routing. *ACM/IEEE Trans. on Networking*, 2002.

Table 2: Performance of the models on `Garr` and `Waxman` instances

instance	model	time		nodes		failed	rate
		avg	max	avg	max		
Garr199901	B	0.01	0.03	0.02	0.55	0.25	0.87
	S	0.00	0.02	0.03	0.63	0.12	0.31
	W	0.00	0.02	0.01	0.37	0.13	0.18
Garr199904	B	0.01	0.04	0.01	0.58	0.14	0.73
	S	0.00	0.02	0.00	0.14	0.12	0.52
	W	0.00	0.02	0.00	0.11	0.12	0.12
Garr199905	B	0.01	0.05	0.01	0.98	0.16	0.74
	S	0.00	0.02	0.00	0.24	0.13	0.53
	W	0.00	0.02	0.00	0.15	0.13	0.13
Garr200109	B	0.01	0.04	0.01	0.58	0.17	0.72
	S	0.00	0.02	0.00	0.24	0.15	0.51
	W	0.00	0.02	0.00	0.18	0.15	0.14
Garr200112	B	0.01	0.05	0.00	0.49	0.16	0.73
	S	0.00	0.02	0.00	0.02	0.14	0.52
	W	0.00	0.02	0.00	0.21	0.14	0.14
Garr200404	B	0.01	0.04	0.01	0.73	0.13	0.74
	S	0.00	0.02	0.00	0.13	0.12	0.54
	W	0.00	0.02	0.00	0.07	0.12	0.12
Garr200908	B	0.05	0.44	0.13	9.78	0.08	0.70
	S	0.00	0.02	0.00	0.98	0.02	0.31
	W	0.00	0.02	0.00	1.16	0.03	0.03
Garr200909	B	0.05	0.39	0.18	16.15	0.08	0.70
	S	0.00	0.02	0.00	1.12	0.03	0.30
	W	0.00	0.02	0.00	1.28	0.03	0.03
Garr200912	B	0.05	0.40	0.17	12.31	0.10	0.70
	S	0.00	0.02	0.00	1.27	0.02	0.30
	W	0.00	0.02	0.00	1.64	0.03	0.03
Garr201001	B	0.05	0.39	0.18	16.18	0.09	0.70
	S	0.00	0.02	0.00	1.25	0.02	0.30
	W	0.00	0.02	0.00	1.60	0.03	0.03
w1-100	B	0.27	4.85	2.16	35.35	0.28	0.78
	S	0.01	0.04	0.00	0.37	0.00	0.11
	W	0.01	0.07	0.00	0.17	0.00	0.00
w1-200	B	4.66	95.66	14.27	224.06	0.23	0.79
	S	0.02	0.07	0.00	0.07	0.00	0.12
	W	0.04	0.45	0.01	4.25	0.00	0.00

Table 3: Performance of the models on `Sndlib` instances

instance	model	time		nodes		failed	rate
		avg	max	avg	max		
abilene	B	0.00	0.01	0.00	0.08	0.23	0.78
	S	0.00	0.01	0.00	0.00	0.07	0.46
	W	0.00	0.01	0.00	0.00	0.06	0.16
atlanta	B	0.00	0.01	0.01	0.49	0.18	0.82
	S	0.00	0.01	0.00	0.03	0.05	0.34
	W	0.00	0.01	0.00	0.04	0.04	0.07
cost266	B	0.00	0.03	0.01	0.82	0.10	0.80
	S	0.00	0.01	0.00	0.03	0.02	0.39
	W	0.00	0.01	0.00	0.01	0.02	0.03
dfn-bwin	B	0.00	0.01	0.00	0.00	0.17	0.92
	S	0.00	0.01	0.00	0.00	0.00	0.81
	W	0.00	0.01	0.00	0.00	0.00	0.00
dfn-gwin	B	0.00	0.01	0.00	0.00	0.21	0.91
	S	0.00	0.01	0.00	0.00	0.02	0.66
	W	0.00	0.01	0.00	0.00	0.02	0.02
di-yuan	B	0.01	0.06	0.53	15.46	0.33	0.84
	S	0.00	0.01	0.00	0.22	0.04	0.41
	W	0.00	0.01	0.01	0.20	0.17	0.18
france	B	0.00	0.02	0.01	0.40	0.19	0.84
	S	0.00	0.01	0.00	0.00	0.03	0.35
	W	0.00	0.01	0.00	0.01	0.02	0.04
geant	B	0.00	0.02	0.02	0.67	0.18	0.84
	S	0.00	0.01	0.00	0.09	0.05	0.41
	W	0.00	0.01	0.00	0.06	0.05	0.07
germany50	B	0.01	0.09	0.07	4.60	0.16	0.84
	S	0.00	0.01	0.00	0.23	0.04	0.37
	W	0.00	0.01	0.00	0.02	0.04	0.06
giul39	B	0.07	0.78	0.66	36.50	0.24	0.87
	S	0.01	0.03	0.00	1.32	0.00	0.15
	W	0.01	0.04	0.00	0.93	0.00	0.01

Table 4: Performance of the models on `Sndlib` instances, cont'd

instance	model	time		nodes		failed	rate
		avg	max	avg	max		
india35	B	0.01	0.06	0.02	1.33	0.19	0.85
	S	0.00	0.01	0.00	0.31	0.04	0.35
	W	0.00	0.01	0.00	0.02	0.04	0.07
janos-us	B	0.04	0.31	0.29	11.56	0.17	0.81
	S	0.00	0.03	0.03	2.96	0.01	0.17
	W	0.00	0.03	0.01	1.83	0.02	0.02
janos-us-ca	B	0.07	0.59	0.66	17.72	0.19	0.83
	S	0.00	0.04	0.02	3.28	0.01	0.17
	W	0.01	0.05	0.01	1.99	0.01	0.02
newyork	B	0.00	0.02	0.02	0.96	0.23	0.88
	S	0.00	0.01	0.00	0.20	0.08	0.42
	W	0.00	0.01	0.00	0.05	0.08	0.12
nobel-eu	B	0.00	0.04	0.04	1.85	0.18	0.83
	S	0.00	0.01	0.01	0.28	0.06	0.39
	W	0.00	0.01	0.00	0.00	0.07	0.10
nobel-germany	B	0.00	0.02	0.01	0.45	0.20	0.80
	S	0.00	0.01	0.00	0.00	0.05	0.45
	W	0.00	0.01	0.00	0.00	0.04	0.07
nobel-us	B	0.00	0.01	0.00	0.00	0.08	0.75
	S	0.00	0.01	0.00	0.02	0.00	0.47
	W	0.00	0.01	0.00	0.01	0.00	0.01
norway	B	0.01	0.10	0.12	4.24	0.27	0.74
	S	0.00	0.02	0.02	1.22	0.05	0.22
	W	0.00	0.02	0.01	0.64	0.06	0.08
pdh	B	0.01	0.04	0.28	4.06	0.26	0.74
	S	0.00	0.01	0.05	1.78	0.08	0.28
	W	0.00	0.01	0.01	0.37	0.11	0.11

Table 5: Performance of the models and schedulers on individual instances

instance	model	time		nodes		failed	rate
		avg	max	avg	max		
w1-200	B-SRP	3.65	110.99	12.02	161.99	0.00	0.80
	B-GB	4.46	83.81	7.84	233.32	0.84	1.00
	B-WRP	6.06	110.28	18.86	262.87	0.00	0.64
	B-FB	4.26	65.72	11.94	247.31	0.69	0.93
	S-SRP	0.02	0.07	0.00	0.08	0.00	0.12
	S-WRP	0.02	0.07	0.00	0.08	0.00	0.12
	S-FB	0.02	0.08	0.00	0.06	0.00	0.12
	W-SRP	0.04	0.44	0.01	4.35	0.00	0.00
	W-WRP	0.04	0.45	0.01	4.35	0.00	0.00
	W-FB	0.04	0.47	0.01	4.07	0.00	0.00
atlanta	B-SRP	0.00	0.01	0.00	0.01	0.05	0.85
	B-GB	0.00	0.01	0.00	0.00	0.97	1.00
	B-WRP	0.00	0.02	0.02	0.90	0.04	0.73
	B-FB	0.00	0.01	0.01	0.55	0.43	0.89
	S-SRP	0.00	0.01	0.00	0.04	0.04	0.33
	S-WRP	0.00	0.01	0.00	0.04	0.04	0.33
	S-FB	0.00	0.01	0.00	0.01	0.07	0.35
	W-SRP	0.00	0.01	0.00	0.05	0.03	0.06
	W-WRP	0.00	0.01	0.00	0.05	0.03	0.06
	W-FB	0.00	0.01	0.00	0.04	0.06	0.08
Garr199904	B-SRP	0.01	0.03	0.00	0.00	0.12	0.98
	B-GB	0.01	0.03	0.00	0.00	1.00	1.00
	B-WRP	0.01	0.03	0.00	0.14	0.12	0.55
	B-FB	0.01	0.07	0.03	1.60	0.17	0.66
	S-SRP	0.00	0.01	0.00	0.11	0.12	0.52
	S-WRP	0.00	0.01	0.00	0.11	0.12	0.52
	S-FB	0.00	0.02	0.00	0.20	0.12	0.52
	W-SRP	0.00	0.02	0.00	0.12	0.12	0.12
	W-WRP	0.00	0.02	0.00	0.12	0.12	0.12
	W-FB	0.00	0.02	0.00	0.10	0.12	0.12