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

Antonio Frangioni, Laura Galli, Giovanni Stea

PII: S0305-0548(16)30308-2
DOI: 10.1016/j.cor.2016.12.009
Reference: CAOR 4147

To appear in: *Computers and Operations Research*

Received date: 29 December 2015
Revised date: 7 December 2016
Accepted date: 7 December 2016

Please cite this article as: Antonio Frangioni, Laura Galli, Giovanni Stea, Delay-constrained Routing Problems: Accurate Scheduling Models and Admission Control, *Computers and Operations Research* (2016), doi: 10.1016/j.cor.2016.12.009

Highlights

- We extend previously available models of delay-constrained routing problems.
- We account for the latency formulae of the most relevant classes of GPS-derived schedulers from the literature.
- We explicitly introduce the concept of admission control constraints, and show how to implement them for all classes of schedulers.
- We show how to model the difference between reserved and guaranteed rates taking into account admission control.
- We prove that the all models are MI-SOCP and computationally viable for real-life communication networks.

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

Antonio Frangioni[a], Laura Galli[a], Giovanni Stea[b]

[a] *Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy. E-mail:* `frangio@di.unipi.it`, `laura.galli@unipi.it`
[b] *Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Largo Lucio Lazzarino 1, 56122 Pisa, Italy. E-mail:* `giovanni.stea@unipi.it`

Abstract

As shown in [1], the problem of routing a flow subject to a worst-case end-to-end delay constraint in a packed-based network can be formulated as a Mixed-Integer Second-Order Cone Program, and solved with general-purpose tools in real time on realistic instances. However, that result only holds for one particular class of packet schedulers, *Strictly Rate-Proportional* ones, and implicitly considering each link to be fully loaded, so that the *reserved rate* of a flow coincides with its *guaranteed rate*. These assumptions make latency expressions simpler, and enforce perfect isolation between flows, i.e., admitting a new flow cannot increase the delay of existing ones. Other commonplace schedulers both yield more complex latency formulæ and do not enforce flow isolation. 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 unloaded. In this paper we extend the result to other classes of schedulers and to a more accurate representation of the latency, showing that, even when admission control needs to be factored in, the problem is still efficiently solvable for realistic instances, provided that the right modeling choices are made.

Keywords: Routing problems, maximum delay constraints, scheduling algorithms, admission control, Second-Order Cone Programs, Perspective Reformulation
2010 MSC: 90B18, 90C35, 90C90

1. Introduction

The Internet already supports applications that require stringent guarantees on end-to-end delays (voice/video streaming, remote operation of industrial/medical tools, etc.). Obtaining *Quality of Service* (QoS) guarantees for a packet flow, such as a maximum delay, is thus a crucial problem, which is made nontrivial by the packed-based nature of the infrastructure. *QoS routing* is the practice of computing network paths where a suitable QoS can be guaranteed, which gives rise

*Corresponding author
Email address: `frangio@di.unipi.it` (Antonio Frangioni)

to Constrained Shortest Path (CSP) problems. CSPs having a single end-to-end constraint which is an additive or multiplicative *concave* function of per-link metrics admit polynomial solution algorithms, while CSPs with two or more constraints are $\mathcal{NP}$-hard (cf. [1] and the references therein). Due to the typically strict requirements on the time to deliver the solution in practice (say, some 10s or 100s of milliseconds), approximate approaches are normally employed to solve them (e.g., [2, 3, 4]). Furthermore, rather simplified network models have been traditionally employed where the relevant QoS parameters, say link delays, are considered statically known and additive. This neglects *queueing*, i.e., the delay due to the fact that the same link is shared by different flows, whose packets are transmitted sequentially. Queueing delays depend on the *packet schedulers* employed to arbitrate the flows.

A well-known paradigm for QoS scheduling is Generalized Processor Sharing (GPS) [5], that defines an ideal reference system which serves backlogged flows simultaneously at a rate proportional to their *weight*. If flow weights are chosen equal to their *reserved rates*, and their sum does not exceed the link capacity, then GPS guarantees that the flows' *guaranteed rates* will be at least as large as the reserved ones. This allows per-link and end-to-end Worst-case Delay (WCD) bounds to be computed if the traffic arrival rate at the source is constrained. Two practical implementations of GPS have been proposed, namely *Packet-by-packet Generalized Processor Sharing (PGPS)* [5] and *Worst-case Fair Weighted Fair Queueing (WF2Q)* [6]. Both exhibit tight guarantees on the *latency*, i.e., the worst-case scheduling delay at a link, which is—barring a small additive constant—inversely proportional to the guaranteed rate, thereby earning them the moniker of *Strictly Rate-Proportional* (SRP) schedulers. Since a flow's WCD depends on the guaranteed rates along its path, QoS routing problems with WCD constraints can easily be defined assuming SRP schedulers in the network. For instance, [7, 8] show that the problem of finding a path with a pre-specified WCD is $\mathcal{NP}$-hard in general, unless the *same rate* is reserved at each link. Recently, [1] showed that, nonetheless, optimal solutions can be found in split-second times for realistic-sized networks even allowing different rates for each link, and that this leads to sizable performance gains in terms of flow blocking probability [9].

Unfortunately, the implementation of SRP schedulers is rather complex, which is a downside on high-speed links and/or with many simultaneous flows. In the last two decades, several schedulers have been devised which exhibit different trade-offs between latency and implementation cost, some having made their way into commercial hardware [10]. At one end of the spectrum we find GPS approximations based on *flow grouping* [11, 12], and at the other end lie *frame-based* algorithms such as Deficit Round Robin [13] and its derivatives [14, 15, 10]. Both are simpler, but exhibit looser latency guarantees 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, [16], shows that non-uniform rate allocation *given a pre-specified routing plan* achieves better network utilization than uniform rate allocation in the presence of WCD 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 blocking probability).

Furthermore, all previous works—including [1, 9]—have resorted to simplifying the latency formulæ by assuming that the *guaranteed* rate of a flow is equal to its *reserved* rate. This *bound assumption* is safe, in that the reserved rate is always no larger than the guaranteed one, but it leads to over-estimating the WCD experienced by a flow, and therefore to a more conservative resource allocation than 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 WCD constraint, determine a feasible path and a feasible rate reservation on each link (if there exists one) minimizing the total reservation cost and ensuring that existing flows still satisfy their WCD 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 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.

This paper is organized as follows. In Section 2 we present our system model and hypotheses. In Sections 3, 4 and 5 we discuss models for the three main classes of latency formulæ—respectively, Strictly Rate-Proportional (and their Group-Based approximations), Weakly Rate-Proportional and Frame-Based ones—under different assumptions on the description of reserved and guaranteed rates. In Section 6 we report computational results which show the relative efficiency (and, partly, effectiveness) of the various models on real networks with realistic traffic data. Finally, in Section 7 we draw some conclusions.

2. System model

We are given a computer network represented by a directed graph $G = (N, A)$, with $n = |N|$ and $m = |A|$. Nodes are switching elements (e.g., routers), and arcs are the links interconnecting them. Henceforth, delays are in seconds, packet lengths are in bits, and rates and link speeds are in bits per second. Each node $i \in N$ is characterized by a fixed *node delay* n_i. Each arc $(i, j) \in A$ is characterized by a fixed *link delay* l_{ij}, a *physical link speed* w_{ij}, and the *maximum transmit* *unit* (MTU) L (assumed to be constant for simplicity). A set Q of flows is already present in the network. Each $q \in Q$ is characterized by a fixed path in G (which, for notational simplicity, we will denote by q as well), fixed reserved rates r^q_{ij} for all $(i,j) \in q$, an upper bound on the tolerable WCD—called its *deadline*—δ^q, and a *leaky-bucket arrival-curve constraint*. That is, if $F(t)$ denotes the number of bits of the flow injected at the source in $[0,t)$, $F(t+\tau) - F(t) \leq \sigma^q + \rho^q\tau$ has to hold for all t and $\tau \geq 0$, where σ^q and ρ^q are the *burst* and the *rate* of the flow, respectively.

We now introduce the *Admissible Delay-Constrained Shortest Path* (ADCSP) problem: given the current state Q of the network, the cost f_{ij} of reserving one unit of capacity on (i,j), and the data describing one "new" flow to be routed in G (its endpoints s and d, burst σ and rate ρ, and deadline δ), find one feasible s-d path p and a feasible reservation of capacity at each of its arcs—if any exist—so that the flow can be routed along p and *both the new flow and all the existing ones* meet their deadline, at the minimum possible reservation cost for the new flow. ADCSP requires one to compute the WCD of a flow, which depends on several factors:

1. the selected *routing* for the flow, i.e., the s-d path p in G;
2. the *reserved rate* $r_{ij} \in [0, w_{ij}]$ for each arc $(i,j) \in p$;
3. the latency guarantees of the schedulers employed to share the output links' bandwidth among the flows (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. the paths and reserved rates of all the other flows $q \in Q$.

In the following, we will denote by $P(i,j) = \{\, q \,:\, (i,j) \in q \,\} \subseteq Q$ the set of existing paths (*not* counting the one just to be routed) traversing arc (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, in general the WCD rather depends on the *guaranteed rate* g_{ij} obtained by the flow on each $(i,j) \in p$. For all the fair-queueing schedulers that we will examine, the guaranteed rate is at least as large as the reserved rate. In fact, under the assumption that the arc is not over-provisioned

$$r_{ij} \leq w_{ij} - \bar{r}_{ij} \quad , \tag{1}$$

where $\bar{r}_{ij} = \sum_{q \in P(i,j)} r^q_{ij}$ (≥ 0) is the total reserved rate of all the *other* flows at link (i,j), the guaranteed rates are given by the expression

$$g_{ij} = (w_{ij} r_{ij})/(\bar{r}_{ij} + r_{ij}) \quad . \tag{2}$$

It is easy to see that $g_{ij} = w_{ij}$ when $\bar{r}_{ij} = 0$ ($\equiv (i,j) \in A'$), i.e., the arc is "completely unloaded" and the new flow is the only one traversing it. Conversely, $g_{ij} = r_{ij}$ when $\bar{r}_{ij} + r_{ij} = w_{ij}$, i.e., the arc is “completely loaded”. In order for the WCD to be finite, the minimum guaranteed rate among all links of p must be at least as large as the traffic injection rate of the flow, i.e.,

$$g_{ij} \geq \rho \qquad \forall (i,j) \in p \ . \tag{3}$$

If (3) is satisfied, the general form of the WCD of path p is

$$\frac{\sigma}{\min\{\, g_{ij} \,:\, (i,j) \in p\}} + \sum_{(i,j)\in p} \big(\theta_{ij} + l_{ij} + n_i \big) \ , \tag{4}$$

where θ_{ij} is the *link latency* experienced by the flow on path p when traversing the arc (i,j), i.e. the maximum scheduling delay that its head-of-line packets may undergo due to the presence of competing flows [17]. Note that θ_{ij} does not incorporate the fixed node traversal and link propagation delays, which are in fact separately considered in (4). The exact form of θ_{ij} depends on the scheduling algorithm employed at the link. In all the previous developments in the literature [1, 7, 8, 9, 16], a “pessimistic” view has been taken, where one assumes that

$$g_{ij} = r_{ij} \qquad \forall (i,j) \in p \ . \tag{5}$$

In turn, (5) logically implies that $(i,j) \in A''$. We will refer to the delay formulæ obtained under assumption (5) as the *bound* delay estimates, as opposed to the more accurate *worst-case* estimates obtained by eschewing (5) and employing the actual formulæ based on the guaranteed rates. While bound estimates on the WCD are safe, they may be arbitrarily loose. To the best of our knowledge, this is the first work in which worst-case formulæ are studied.

Although many different schedulers have been proposed in the literature, differing regarding several properties, from the point of view of latency most of them fall into the following categories:

- *Strictly Rate-Proportional (SRP)* schedulers, such as PGPS [5] and WF2Q [6], with *worst-case* latency and *bound* latency, respectively, of

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

$$\theta_{ij} = \frac{L}{r_{ij}} + \frac{L}{w_{ij}} \ . \tag{7}$$

  SRP latency can only be achieved by schedulers that require relatively complex implementations [18, 19], which can be a burden at high link speeds, when only a few *ns* are available to make scheduling decisions.

- *Group-based approximations of SRP (GSRP)*, e.g. [11, 12], which group flows according to their reserved rate, at logarithmic intervals, thereby decreasing the complexity but increasing the latency to [11]

$$\theta_{ij} = 3\frac{2^{\lceil \log_2 w_{ij} L / r_{ij} \rceil}}{w_{ij}} + 2\frac{L}{w_{ij}} \ . \tag{8}$$

The complex expression (8) can be shown to satisfy the simpler bounds

$$3\frac{L}{r_{ij}} + 2\frac{L}{w_{ij}} \leq \theta_{ij} \leq 6\frac{L}{r_{ij}} + 2\frac{L}{w_{ij}} \quad . \tag{9}$$

Hence, the latency is still (approximately) rate-proportional, but three to six times larger than (7), even disregarding the constant term, which is twice as large. To the best of our knowledge, worst-case versions of (9) not only are not know, but are considered to be unlikely to exist at all [20], hence only the bound version can be used.

- Schedulers with *Weakly Rate-Proportional* (WRP) latency, e.g., Self-Clocked Fair Queueing [21], whose (worst-case and bound) latency also depends on the *number of flows* $|P(i,j)|$:

$$\theta_{ij} = |P(i,j)|\frac{L}{w_{ij}} + \frac{L}{g_{ij}} \quad , \tag{10}$$

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

  The non-rate-proportional offset in (10)/(11) is larger than in (6)/(7), especially if $|P(i,j)|$ is large; thus increasing the reserved rate may decrease the latency only marginally.

- *Frame-based* (FB) schedulers [13, 14, 15] 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. If the quantum is lower-bounded by the MTU L, the flow requesting the smallest reserved rate must get a quantum equal to the lower bound L, and all the other flows get their quanta scaled accordingly. Thus, the latency, besides the number of flows, also depends not only on the sum $\bar{r}_{ij}$ of the reserved rates of other flows on the link, but also on their minimum $r_{ij}^{min} = \min\{\, r_{ij}^q \,:\, q \in P(i,j)\,\}$ $(\geq \rho)$. Some straightforward algebraic manipulations [22] on the expression reported in [15] give

$$\theta_{ij} = \frac{L}{w_{ij}} \frac{\bar{r}_{ij}}{\min\{\, r_{ij} \,,\, r_{ij}^{min}\,\}} + |P(i,j)|\frac{L}{w_{ij}} + \frac{L}{g_{ij}} \quad , \tag{12}$$

$$\theta_{ij} = \frac{L}{w_{ij}} \frac{w_{ij} - r_{ij}}{\min\{\, r_{ij} \,,\, r_{ij}^{min}\,\}} + |P(i,j)|\frac{L}{w_{ij}} + \frac{L}{r_{ij}} \quad , \tag{13}$$

  for the worst-case and bound versions, respectively. The difference between the two depends on the fact that the bound assumption (5), besides $g_{ij} = r_{ij}$, also implies that $\bar{r}_{ij} + r_{ij} = w_{ij}$. Note that the difference depends on r_{ij}, i.e., on the *reserved* rates rather than on the guaranteed ones. Some FB schedulers [14, 15] have latency expressions similar to (12)/(13), save for a multiplying constant $\kappa > 1$ at the denominator of the first term. As the impact of this change on the models is straightforward, we will limit our presentation to (12)/(13).

Comparing formulæ (6) and (10) shows that SRP schedulers have smaller latency than WRP ones. Remarkably, for the bound ones (7) and (11) there would seem to be an exception for the case of the “completely unloaded” arcs in A', but this 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 (10)/(11) with (12)/(13) also shows that the WRP latency is always smaller than or equal to that of FB.

In general, the link latency expression of all schedulers depends on the routing and reservation choices for *all* the other flows. In the bound SRP case (7) this is only due to (1); hence, provided that links are not oversubscribed, the new flow p can be routed without the others being affected. However, in all other cases routing p in the network can (and does) affect the delay of any existing flow q that shares at least one arc (i, j) with p, 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.

Since we assume that the routing of the new flow is only possible if existing flows remain delay-feasible without their routing/rates decisions be changed, 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 has focused on the bound formula (7) for SRP schedulers, for which admission control is "automatic" (note that the same also holds for GSRP schedulers). In the next sections we will show that this is not the case with worst-case formulæ and/or WRP and FB schedulers.

3. Strictly Rate Proportional and Group-based Schedulers

In this section we describe ADCSP for SRP and GSRP schedulers. We start by presenting a SRP model under the bound assumption (5), which is a modification of that of [1], that will be the basis for all the other ADCSP models. We then discuss how to modify the model to use worst-case formulæ instead. 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. As these two aspects can be partly separated, i.e., one can write models where only one of the two modifications is performed, we discuss them separately.

3.1. SRP with bound assumption

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 $x_{ij} = 0$. With these, one can define the following formulation for the problem, under the bound SRP formula (7):

$$\min \textstyle\sum_{(i,j)\in A} f_{ij} r_{ij} \tag{14}$$

$$\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = \begin{cases} -1 & \text{if } i = s \\ 1 & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \qquad i \in N \tag{15}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A \tag{16}$$

$$\sigma t + \textstyle\sum_{(i,j)\in A} \big[\, \theta_{ij} + (l_{ij} + n_i) x_{ij} \,\big] \leq \delta \tag{17}$$

$$s_{ij} r_{ij} \geq x_{ij}^2 \;\;,\;\; s_{ij} \geq 0 \qquad (i,j) \in A \tag{18}$$

$$t \geq s_{ij} \qquad (i,j) \in A \tag{19}$$

$$\rho x_{ij} \leq r_{ij} \leq (w_{ij} - \bar{r}_{ij}) x_{ij} \qquad (i,j) \in A \tag{20}$$

$$\theta_{ij} = L s_{ij} + (L/w_{ij}) x_{ij} \qquad (i,j) \in A \tag{21}$$

The objective function (14) represents the total reservation cost. Most often $f_{ij} = 1$, which can be useful for algorithmic purposes [1]. Note that (14) could be trivially generalized to the case where there is a given fixed cost for selecting arcs. The standard flow conservation constraints (15), together with the integrality ones (16), ensure that the x_{ij} variables represent a s-d path. Constraint (17) imposes the end-to-end delay restriction, implementing (4) provided that appropriate conditions are imposed, so that the variables θ_{ij} correctly represent the link latencies for the given scheduler, and that the auxiliary variable t represents the term σ/r_{min} in (4). The (rotated) Second-Order Cone (SOCP) constraints (18) ensure that

$$s_{ij} = \begin{cases} 1/r_{ij} & \text{if } x_{ij} = 1 \\ 0 & \text{if } x_{ij} = 0 \end{cases} .$$

These employ the *Perspective Reformulation technique* [23, 24] to ensure that the resulting formulation is "tight", which results in significantly improved performances w.r.t. standard "big-M" formulations [1]. The constraints (19) impose that $t = \sigma/r_{min}$, based on the fact that

$$t = \max\{\, s_{ij} (= 1/r_{ij}) \;:\; (i,j) \in p \,\} = 1/\min\{\, r_{ij} \;:\; (i,j) \in p \,\} = 1/r_{min} \;\; . \tag{22}$$

This part of the formulation is different from the one originally presented in [1], where an explicit variable r_{min} was introduced with the appropriate constraints. The version presented here has been found to be computationally convenient in practice, at least on out test bed. The details are omitted for space reasons, but can be found in [22]. Finally, constraints (20) enforce the semi-continuous nature of reserve rate variables r_{ij}, i.e., $x_{ij} = 0 \Longrightarrow r_{ij} = 0$ and $x_{ij} = 1 \Longrightarrow r_{ij} \in [\rho, w_{ij} - \bar{r}_{ij}]$.

All the constraints discussed so far are independent of the specific link latency formula, and therefore of the exployed scheduler. The specific (7) is implemented in (21). Clearly, the θ_{ij}

variables are useless in the formulation: substituting them into (17) would just yield

$$\sigma t + \sum_{(i,j)\in A} \big[\, L s_{ij} + (L/w_{ij} + l_{ij} + n_i) x_{ij} \,\big] \leq \delta \ . \tag{23}$$

This is not particularly relevant, not even computationally: typically, the preprocessor of any general-purpose solver will form (23) out of (17) and (21) automatically anyway. Because the second term in the sum in (23) will be common to many cases, for the sake of notational simplicity we define $\bar{l}_{ij} = L/w_{ij} + l_{ij} + n_i$. The formulation (14)–(21) 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 [1] to be efficient and effective for solving realistic instances.

3.2. SRP with guaranteed rates in the link latency

To implement the worst-case link latency formula (6), one just has to use (2) to get

$$\theta_{ij} = \frac{L}{w_{ij}} + \begin{cases} (L/w_{ij})/(\bar{r}_{ij}/r_{ij} + 1) & \text{if } (i,j) \in A'' \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

Thus, employing (24) instead of (7) in the models is relatively straightforward. Indeed, the "empty" arcs have constant latency (in fact, $\bar{r}_{ij} = 0 \Longrightarrow g_{ij} = w_{ij}$), and the others, besides yet another constant, have the same "$1/r_{ij}$" form, with just a different scaling factor. Hence, the optimization problem is largely unaffected: one only has to replace (21) with

$$\theta_{ij} = \frac{L}{w_{ij}} + \begin{cases} (L\bar{r}_{ij}/w_{ij}) s_{ij} + (L/w_{ij}) x_{ij} & \text{if } (i,j) \in A'' \\ 0 & \text{otherwise} \end{cases} \quad , \tag{25}$$

which is in no way significantly more complex. In fact, it is somewhat less so, since for arcs in A', the variables θ_{ij} and s_{ij} and the corresponding conic constraints (18) are not needed. Of course, the θ_{ij} variables could be eliminated similarly to (23).

However, removing assumption (5) also implies that the delay depends on the other flows via the term $\bar{r}_{ij}$ in (24), and therefore admission control is required. Indeed, consider an existing flow q and some $(i,j) \in q$ such that $r_{ij} > 0$ ($\Longrightarrow x_{ij} = 1$). Note that this means that, "from the viewpoint of q", $P(i,j)$ is nonempty because it contains at least the new flow, i.e., $(i,j) \in A''$. The arrival of the new flow then affects the delay of q because

$$g^q_{ij} = \frac{r^q_{ij} w_{ij}}{\bar{r}_{ij} + r_{ij}} \quad \Longrightarrow \quad \theta^q_{ij} = \frac{L}{w_{ij}} + \frac{L}{g^q_{ij}} = \frac{L}{w_{ij}} \left(\frac{\bar{r}_{ij} + r_{ij}}{r^q_{ij}} + 1 \right) = \frac{L r_{ij}}{w_{ij} r^q_{ij}} + \frac{L}{w_{ij}} \left(\frac{\bar{r}_{ij}}{r^q_{ij}} + 1 \right) \ . \tag{26}$$

This means that the increase in latency due to the new flow passing through (i,j) is

$$\Delta\theta^q_{ij} = \frac{L\, r_{ij}}{w_{ij}\, r^q_{ij}} + \begin{cases} 0 & \text{if } |P(i,j) \setminus \{q\}| > 0 \\ L/w_{ij} & \text{otherwise} \end{cases} \quad .$$

In fact, the rightmost addendum in the last part of (26) was the previous delay (for $r_{ij} = 0$) if $P(i,j) \setminus \{q\} \neq \emptyset$, i.e., q was not the only flow on that arc. If $P(i,j) = \{q\}$ instead, the delay experienced by q was originally L/w_{ij}, which then becomes $2L/w_{ij}$ ($\bar{r}_{ij} = r^q_{ij}$) plus the part depending on r_{ij}. Hence, the "+1" in the last addendum of (26) has to be counted towards $\Delta\theta^q_{ij}$. 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, and define the *delay slack*

$$\bar{\delta}^q = \delta^q - \tfrac{\sigma^q}{r^q_{min}} - \textstyle\sum_{(i,j)\in q} \bar{l}_{ij} - \sum_{(i,j)\in q''} \frac{L\bar{r}_{ij}}{w_{ij} r^q_{ij}} \; (\geq 0) \quad , \tag{27}$$

i.e., the maximum extra delay that q can tolerate without violating the corresponding WCD constraint. One can then ensure that the delay of all flows remains feasible by simply adding to the model the linear admission control constraints

$$\textstyle\sum_{(i,j)\in q} \frac{L}{w_{ij} r^q_{ij}} r_{ij} + \sum_{(i,j)\in q'} \frac{L}{w_{ij}} x_{ij} \leq \bar{\delta}^q \qquad q \in Q \quad . \tag{28}$$

Thus, removing assumption (5) for SRP schedulers hardly changes the model, save for adding as many linear constraints as the flows currently in the network. The overall SRP model with worst-case link latency just reads (14)–(20), (25), (28).

3.3. SRP with guaranteed rates in the end-to-end delay

So far we assumed that guaranteed rates only affect the link latency expression. However, guaranteed rates are also involved in both the *lower bound* constraints (3) and the *minimum rate* term in (4). Constraint (20) becomes

$$\frac{\rho\, \bar{r}_{ij}}{w_{ij} - \rho} x_{ij} \leq r_{ij} \leq (w_{ij} - \bar{r}_{ij}) x_{ij} \qquad (i,j) \in A \quad , \tag{29}$$

obtained by just plugging (2) in (3). Note that the upper bound on r_{ij} is not affected, because (1) is related to the *reserved* rate instead of the *guaranteed* one. As far as (4) is concerned, the change is that t in (17) now has to represent $1/g^{min}_{ij}$ instead of $1/r^{min}_{ij}$. Fortunately, one can readily extend the approach of (22): just use (2) to get

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

We can now exploit the fact that $s_{ij} = 1/r_{ij}$ and simply replace (19) with

$$t \geq (\bar{r}_{ij}/w_{ij}) s_{ij} + (1/w_{ij}) x_{ij} \qquad (i,j) \in A \quad , \tag{30}$$

irrespectively of the constraints on θ_{ij}, i.e., of the choice of the scheduler.

However, (4) is not only relevant for (17): it is also the basis of the admission control constraints, which are needed when using worst-case formulæ. The impact is limited, though: in the delay slack (27), one has to separate the term "σ^q/r^q_{min}" from the rest, since this now has to be "σ^q/g^q_{min}". That is, one has to consider

$$\bar{\delta}^q = \tilde{\delta}^q - \sigma^q/r^q_{min} \quad \text{where} \quad \tilde{\delta}^q = \delta^q - \textstyle\sum_{(i,j)\in q} \bar{l}_{ij} - \sum_{(i,j)\in q''} (L\bar{r}_{ij})/(w_{ij}r^q_{ij}) \, (\geq 0) \ . \tag{31}$$

Indeed, since g^q_{min} depends on all the other flows, comprised the new one currently being routed, the term σ^q/g^q_{min} is no longer independent of the variables of the problem. However, this can be dealt with similarly to (30):

$$\frac{1}{g^q_{min}} = \max\left\{ \frac{1}{g^q_{ij}} \,:\, (i,j)\in q \right\} = \max\left\{ \frac{\bar{r}_{ij} + r_{ij}}{w_{ij}r^q_{ij}} \,:\, (i,j)\in q \right\} .$$

Hence, introducing extra variables t^q for all $q \in Q$, one can rewrite (28) as

$$\textstyle\sum_{(i,j)\in q} \frac{L}{w_{ij}r^q_{ij}} r_{ij} + \sum_{(i,j)\in q'} \frac{L}{w_{ij}} x_{ij} + t^q\sigma^q \leq \tilde{\delta}^q \qquad q \in Q \tag{32}$$

$$(w_{ij}r^q_{ij})t^q \geq \bar{r}_{ij} + r_{ij} \quad (i,j)\in q \ , \quad q \in Q \ . \tag{33}$$

In summary, the SRP model where guaranteed rates are used both in the link latency formulæ and in defining the end-to-end delay (and therefore the admission control constraints) reads (14)–(18), (30), (29), (25), (32)–(33).

3.4. Group-based models

The extension of (14)–(21) for GSRP is straightforward when using the approximation (9). 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 (21) with

$$\theta_{ij} = 3Ls_{ij} + (2L/w_{ij})x_{ij} \ . \tag{34}$$

Clearly, this does not guarantee that the obtained solution is actually delay-feasible. Yet, as our experiments will show, even this less conservative choice leads to significantly more costly solutions than the others, while its computational cost should reasonably be very comparable to that of the more conservative approach. As already mentioned, no worst-case delay formulæ of GSRP have been devised yet, and it is deemed very unlikely that they could be in a future. Hence, no worst-case versions of ADCSP for GSRP can be defined. As an advantage, no admission control mechanism is required in this case, save for testing (1). One could consider employing the more accurate version (8) instead of (9), but this would significantly complicate the mathematical model. In light of the obtained computational results (cf. §6) we have chosen not to pursue this approach in this work.

4. Weakly Rate Proportional Schedulers

We now study the latency models (6)/(11). Similarly to the previous Section, we start with the bound model (11), and then extend it to the worst-case one (6).

4.1. WRP with bound assumption

The extension of the base model is straightforward: just replace (21) with

$$\theta_{ij} = Ls_{ij} + (L/w_{ij})|P(i,j)|x_{ij} \quad . \tag{35}$$

Because $|P(i,j)|$ is a constant, this has no impact on the shape of the optimization model. However, admission control is required, even for the bound case, because the delay of an existing flow q is increased on all the arcs used by the new flow, i.e., where $x_{ij} = 1$, since $|P(i,j)|$ increases there. In order to tackle this, one just has to define the delay slack of q as

$$\bar{\delta}^q = \delta^q - \frac{\sigma^q}{r^q_{min}} - \sum_{(i,j)\in q}\left(\frac{L}{r^q_{ij}} + (|P(i,j)|-1)\frac{L}{w_{ij}} + l_{ij} + n_i\right) \quad . \tag{36}$$

The term $|P(i,j)| - 1$ in (36) is due to the fact that "$|P(i,j)|$" in (11) does *not* count q, as it represents the network state *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 $|P(i,j)| - 1 \geq 0$). It is now sufficient to add the simple admission control constraint

$$\sum_{(i,j)\in q}(L/w_{ij})x_{ij} \leq \bar{\delta}^q \qquad q \in Q \quad , \tag{37}$$

i.e., (28) with no need to distinguish between q' and q''. Once again admission control only comes at the cost of one linear constraint for each existing flow. The bound WRP model simply reads (14)–(20), (35), (37).

4.2. WRP with guaranteed rates

Also for WRP schedulers, removing assumption (5) from the link latency formula leads to a model hardly more complex: using (2) in (10) and $s_{ij} = 1/r_{ij}$ yields

$$\theta_{ij} = (L\bar{r}_{ij}/w_{ij})s_{ij} + (L/w_{ij})(|P(i,j)|+1)x_{ij} \quad . \tag{38}$$

Note that the coefficient of s_{ij} in (38) is zero when $\bar{r}_{ij}$ is (i.e., $(i,j) \in A'$). Thus, some of the auxiliary variables s_{ij} and the corresponding conic constraints (18) can actually be avoided.

As far as admission control is concerned, working as in (26), it is easy to see that the increase in latency for an existing flow q due to the new flow traversing $(i,j) \in q$ is

$$\Delta\theta^q_{ij} = (L/w_{ij})\left(r_{ij}/r^q_{ij} + 1\right) \quad . \tag{39}$$

It is then only necessary to define the delay slack as

$$\bar{\delta}^q = \delta^q - \tfrac{\sigma^q}{r^q_{min}} - \textstyle\sum_{(i,j)\in q} \left[\tfrac{L}{w_{ij}} \left(\tfrac{\bar{r}_{ij}}{r^q_{ij}} + |P(i,j)| - 1 \right) + l_{ij} + n_i \right] \ . \tag{40}$$

Note that $P(i,j)$ here contains q, while in (38) $P(i,j)$ 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

$$\textstyle\sum_{(i,j)\in q} (L/w_{ij}) \big(r_{ij}/r^q_{ij} + x_{ij} \big) \leq \bar{\delta}^q \ \ , \tag{41}$$

and the WRP model with guaranteed rates in the link latency reads (14)–(20), (38), (41).

Fortunately, nothing more is required to include guaranteed rates too in the end-to-end delay: starting from (41) the same modifications as in §3.3 work, only provided that $\tilde{\delta}^q$ is defined as in (31) but using (40) as the basic definition instead of (27), i.e., by just removing the term "σ^q/g^q_{min}" from (40). Of course, the modified bound constraints (29) have to be used as well. Hence, the WRP model where guaranteed rates appear in both the link- and the end-to-end delays—and therefore in the admission control constraints—simply reads (14)–(18), (29)–(30), (38), (32)–(33).

5. Frame-Based schedulers

The latency (12) of FB schedulers can be regarded as being composed of that of SRP ones, i.e., (6), plus the "simple" extra term $(L/w_{ij})|P(i,j)|$ found in (11), plus a further "complex" term. Obviously, the analysis will then focus on the impact of the latter.

5.1. FB with bound assumption

Under the bound assumption (5), the "complex" term (when $x_{ij} = 1$) is

$$\frac{L}{w_{ij}} \frac{w_{ij} - r_{ij}}{\min\{\, r_{ij} \, , \, r^{min}_{ij} \,\}} \ \ . \tag{42}$$

It is easy to verify that (42) is *not* a jointly convex function in r_{ij} and all the r^q_{ij} (r^{min}_{ij}). Luckily, the latter are *fixed* in this setting and therefore so is r^{min}_{ij}. This makes it convex in r_{ij}. To realize this, drop the (i,j) index for notational convenience and consider the function

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

Both ϕ_1 and ϕ_2 are convex, hence the only critical point is $r = r^{min}$. However, since $w/r^{min} \geq 1$, $\phi'_1(r) = -w/r^2$ and $\phi'_2(r) = -1/r^{min}$, one has

$$\phi'_1(r^{min}) = -\frac{w}{(r^{min})^2} = -\frac{1}{r^{min}} \left(\frac{w}{r^{min}} \right) \leq -\frac{1}{r^{min}} = \phi'_2(r^{min}) \ \ ,$$

i.e., the derivative is globally non-decreasing, and therefore ϕ is convex. This would suggest using the classical "variable splitting" approach to represent a convex piecewise function [22], but a better representation can be obtained by just observing that not only $\phi_1(r^{min}) = \phi_2(r^{min})$, but also $\phi_1(w) = \phi_2(w)[= 0]$. It is then easy to verify 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,

$$\phi(r) = \max\{\, \phi_1(r) \;,\; \phi_2(r) \,\} \qquad\qquad \forall r \in [\rho, w] \;\; . \tag{43}$$

Thus, by introducing auxiliary variables v_{ij}, one can represent (42) as

$$\theta_{ij} = Ls_{ij} + v_{ij} + (L/w_{ij})|P(i,j)|x_{ij} \qquad (i,j) \in A \tag{44}$$

$$v_{ij} \geq Ls_{ij} - L/w_{ij} \;\;,\;\; v_{ij} \geq (L/r_{ij}^{min})x_{ij} - Lr_{ij}/(w_{ij}r_{ij}^{min}) \;\;,\;\; v_{ij} \geq 0 \qquad (i,j) \in A \tag{45}$$

where, as usual, we have exploited $s_{ij} = 1/r_{ij}$. Note that multiplying the constant term w_{ij}/r_{ij}^{min} of ϕ_2 by x_{ij} in (45) is necessary, because it allows v_{ij} to be 0 when $r_{ij} = x_{ij} = 0$.

Similarly, admission control constraints for FB schedulers are, basically, those of WRP except for the term (42). That is, one can use the same definition of delay slack (36), since (42) depends on the choices made for the new flow, and is therefore not constant. In other words, one can write the admission control constraint as

$$\sum_{(i,j)\in q} \frac{L}{w_{ij}} \left(x_{ij} + \frac{w_{ij} - r_{ij}^q}{\min\{\, r_{ij} \;,\; r_{ij}^{min} \,\}} \right) \leq \bar{\delta}^q \tag{46}$$

with the $\bar{\delta}^q$ of (36). Exploiting the already discussed property (42), the SOCP formulation

$$\textstyle\sum_{(i,j)\in q} (L/w_{ij}) \big(\, x_{ij} + (w_{ij} - r_{ij}^q) z_{ij} \, \big) \leq \bar{\delta}^q \qquad q \in Q \tag{47}$$

$$z_{ij} \geq 1/r_{ij}^{min} \;\;,\;\; z_{ij} \geq s_{ij} \qquad (i,j) \in q \qquad q \in Q \tag{48}$$

is easily seen to properly implement (46). Note that the "$1/r_{ij}^{min}$" term cannot cause any problems here: $(i,j) \in q \implies (i,j) \notin A' \implies r_{ij}^{min} > 0$. It is also important to remark that *neither the* s_{ij} *nor the* z_{ij} *depend on the flow* q. Hence, these can be defined just once for all arcs $(i,j) \in A$, and then used to define the admission control constraints for all the flows. Actually, the z_{ij} variables only need to be defined for $(i,j) \in A''$, i.e., the arcs upon which at least one existing flow is routed. In summary, the FB bound model is (14)–(20), (44)–(45), (47)–(48).

5.2. FB with guaranteed rates

Removing assumption (5) for FB schedulers leads to formulæ similar to the bound case, although with some differences. The FB formula (12) is the same as (38) plus the extra term

$$\frac{L}{w_{ij}} \frac{\bar{r}_{ij}}{\min\{\, r_{ij} \;,\; r_{ij}^{min} \,\}} \;\; , \tag{49}$$

only for $(i,j) \in A''$; this is quite convenient, because $\bar{r}_{ij} = r_{ij}^{min} = 0$ for $(i,j) \in A'$, making (49) ill-defined. Note again that (49) depends on the reserved rates r_{ij}, even though elsewhere in (12) the guaranteed rates g_{ij} appear. Dropping the (i,j) index, the relevant function is

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

Since $\phi_1'(r^{min}) = -1/(r^{min})^2 < 0 = \phi_2'(r^{min})$, then ϕ is convex. Furthermore, (43) holds. Thus, introducing again the auxiliary variables v_{ij}, one can represent (12) by

$$\theta_{ij} = \frac{L}{w_{ij}} + \begin{cases} (L\bar{r}_{ij}/w_{ij})(s_{ij} + v_{ij}) + (L/w_{ij})|P(i,j)|x_{ij} & \text{if } (i,j) \in A'' \\ 0 & \text{otherwise} \end{cases} \qquad (i,j) \in A \qquad (50)$$

$$v_{ij} \geq s_{ij} \;\;,\;\; v_{ij} \geq 1/r_{ij}^{min} \qquad (i,j) \in A'' \qquad (51)$$

where, again, the variables s_{ij} and v_{ij} need not be defined for $(i,j) \in A'$.

Regarding admission control, the latency increase due to the new flow can be seen as being composed of two terms: the one corresponding to the "WRP part" of the latency formula, and the one corresponding to (49). The first gives the linear increment (39) of the delay, but the second gives a nonlinear increment. We can deal with the latter in the usual way: define the delay slack by (40), i.e., disregarding precisely the FB-specific term, and then add the term corresponding to (49) for the flow q. All this leads to

$$\sum_{(i,j)\in q} \frac{L}{w_{ij}} \left(\frac{\bar{r}_{ij}^{-q} + r_{ij}}{\min\{\, r_{ij} \,,\, r_{ij}^{min} \,\}} + \frac{r_{ij}}{r_{ij}^{q}} + x_{ij} \right) \leq \bar{\delta}^q \quad , \qquad (52)$$

where $\bar{r}_{ij}^{-q} = \sum_{h \in P(i,j) \setminus \{q\}} r_{ij}^h$. Indeed, the denominator of (49) is $\bar{r}$, which means "the sum of all the *other* flows in the arc", i.e., barring the one whose delay is being computed. The latter is p in (49), but is q in (52): therefore, r_{ij} has to be included, but r_{ij}^q has not. Conversely, the denominator has to be the minimum of r_{ij}^h for *all* h, including both p and q, so here r_{ij}^q is counted in r_{ij}^{min}. Now, the analysis follows well-established steps: for

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

one has $r \leq r^{min} \Longrightarrow \phi_1 \geq \phi_2$, and therefore (43) holds. Hence, (52) can be expressed as

$$\textstyle\sum_{(i,j)\in q} (L/w_{ij}) \big(\, z_{ij} + r_{ij}/r_{ij}^q + x_{ij} \, \big) \leq \bar{\delta}^q \qquad q \in Q \qquad (53)$$

$$z_{ij} \geq (\bar{r}_{ij} + r_{ij})/r_{ij}^{min} \;\;,\;\; z_{ij} \geq \bar{r}_{ij} s_{ij} + 1 \qquad (i,j) \in q \qquad q \in Q \qquad (54)$$

Again, (53) is (41) plus the extra term necessary to deal with (49), which requires the new variables z_{ij}. These are "shared" among all admission control constraints, and need only be defined for arcs in A''. This is necessary for (54) to work, in that for $(i,j) \in A'$ (54) would give $z_{ij} \geq 1$ even if

$x_{ij} = 0 \Longrightarrow r_{ij} = 0$, whereas $x_{ij} = 0 \Longrightarrow z_{ij} = 0$. The FB model with guaranteed rates in the link latency then reads (14)–(19), (50)–(51), (53)–(54).

As in the WRP case, extending the treatment to incorporate guaranteed rates in the end-to-end delay only requires one to compose already developed pieces. Starting from (53)–(54), the same modifications as in §3.3 work, only provided that $\tilde{\delta}^q$ is defined as in (31) but using (40) as the basic definition instead of (27) (i.e., by just removing the term "σ^q/g^q_{min}" from (40)). Of course, (29) have to be used. In summary, the FB model where guaranteed rates appear in both the link- and end-to-end delay—and therefore in the admission control constraints—simply reads (14)–(18), (29)–(30), (50)–(51), (32)–(33).

6. Computational results

We have shown that a trade-off potentially exists between more accurate models of the different scheduling algorithms and the size and complexity of the corresponding optimization models. The computational side of this trade-off is explored in this section, where we compare the efficiency—and, partly, the effectiveness—of the different MI-SOCP formulations. All the experiments have been performed on a 2.3 Ghz AMD Opteron 6376 with 16Gb RAM, running Ubuntu 12.4. The models were solved by the state-of-the-art, off-the-shelf solver `Cplex 12.6`, called single-threaded via the C API, which basically took all the running time. Unlike in [1], we have purposely refrained from developing ad-hoc approaches for these models, since the aim of this work is to show that one can exploit the flexibility of general-purpose tools to solve many different variants of the problem, and yet attain reasonable efficiency. Of course, either using different solvers (say, targeting the natural nonlinear formulation of the problem rather than the MI-SOCP one) or developing specialized approaches could further improve the solution times.

6.1. Generating the instances

The test instances were generated as in [1], but we summarize the procedure here for ease of reading. We used both real-world and synthetic topologies. Real-world ones are the `GARR` subset [25] of the Internet Topology Zoo [26] and the `SNDlib` topologies [27]. Since these networks all have less than 100 nodes, we also generated two larger random topologies according to the *Waxman* model [28], with $n \in \{100, 200\}$ and density $n/m = 0.4$. Link capacities were chosen among $\{1, 10, 40\}$ Gbps, according to the link's *edge betweeness* [29]. We set $L = 1500$ bytes at all links. Node delays n_i and link delays l_{ij} were set equal to L/w_{ij}. As for the flows, bursts σ were set to $3L$, and rates ρ were taken from a lognormal distribution with $\mu = 0.8$ Gbps and $\sigma^2 = 0.05$ [29]. 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 deadline constraint becomes redundant. Then, δ was chosen uniformly within the interval $[\,\delta_{min}\,,\,\delta_{min}+(\delta_{max}-\delta_{min})\beta\,]$ for a fixed parameter β. We set $\beta = 0.2$, which produces tight deadlines.

We then performed network simulations—the details of which are described in [30]—to produce different ADCSP instances. In the simulations, flows are generated at exponential interarrival times with rate λ, and, when admitted, last for an exponentially distributed time, with a mean $\mu = 1$. Hence, λ determines the network load: the values $\lambda \in 0.1, 1, 10, 100$ were used to achieve different levels of network congestion. Flows have been routed assuming the bound FB schedulers of §5.1 (with admission control), which have the most conservative latency model. Therefore, the corresponding network configurations are guaranteed to satisfy all admission control constraints for the other latency models as well. We sliced the simulation time into 20 time slots, and disregarded the first 10 to ensure that transitory effects were over. Then, the set of active flows at the beginning of each of the subsequent 10 time slots were recorded. All simulations were ran 5 times with different seeds, yielding 50 instances per load value and a grand total of 200 instances per topology.

6.2. Computational results

The first set of results we report concerns 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 (5). We tested three possible modeling variants:

- *bound* (B) models where the assumption (5) is uniformly used;
- *semi-worst-case* (S) models where worst-case latency formulæ are used, but reserved rates are kept in the WCD constraint (cf. e.g. §3.2);
- *worst-case* (W) models where guaranteed rates are uniformly used.

Results are reported in Tables 1, 2, and 3. 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. The columns are divided into three groups, one for each of B, S and W models. In each group of 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 "fail" 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 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 a given model is in terms of network resources usage. Indeed, a more accurate representation of the delay allows the same WCD guarantee to be met with smaller allocated rates, which leaves more headway for subsequent flows to be routed. Similarly, a lower failure rate means that, in a given state, 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. Since we can compute the allocated rate only for non-failed flows, we assigned a relative rate of 1—corresponding to the maximum rate among non-failed flows—to failed ones. We found that not doing so would unduly skew the results towards schedulers that fail more often.

We 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 times. However, the results for the different latency classes were similar enough that aggregating them was still possible. This trades a little accuracy for a lot of table space, making results more readable. Furthermore, more detailed results are provided later on to discuss the finer nuances of the different schedulers' behaviour. For this reason we did not consider GSRP schedulers at this stage: they are defined only for one of the model classes (B), and since their results are (as we shall see) much worse than the others, they would have considerably skewed the aggregated results.

Table 1: Performance of the models on `Garr` instances

	B						W						S					
name	time		nodes		fail	rate	time		nodes		fail	rate	time		nodes		fail	rate
	avg	max	avg	max			avg	max	avg	max			avg	max	avg	max		
199901	0.01	0.03	0.02	0.55	0.25	0.87	0.00	0.02	0.03	0.63	0.12	0.31	0.00	0.02	0.01	0.37	0.13	0.18
199904	0.01	0.04	0.01	0.58	0.14	0.73	0.00	0.02	0.00	0.14	0.12	0.52	0.00	0.02	0.00	0.11	0.12	0.12
199905	0.01	0.05	0.01	0.98	0.16	0.74	0.00	0.02	0.00	0.24	0.13	0.53	0.00	0.02	0.00	0.15	0.13	0.13
200109	0.01	0.04	0.01	0.58	0.17	0.72	0.00	0.02	0.00	0.24	0.15	0.51	0.00	0.02	0.00	0.18	0.15	0.14
200112	0.01	0.05	0.00	0.49	0.16	0.73	0.00	0.02	0.00	0.02	0.14	0.52	0.00	0.02	0.00	0.21	0.14	0.14
200404	0.01	0.04	0.01	0.73	0.13	0.74	0.00	0.02	0.00	0.13	0.12	0.54	0.00	0.02	0.00	0.07	0.12	0.12
200908	0.05	0.44	0.13	9.78	0.08	0.70	0.00	0.02	0.00	0.98	0.02	0.31	0.00	0.02	0.00	1.16	0.03	0.03
200909	0.05	0.39	0.18	16.15	0.08	0.70	0.00	0.02	0.00	1.12	0.03	0.30	0.00	0.02	0.00	1.28	0.03	0.03
200912	0.05	0.40	0.17	12.31	0.10	0.70	0.00	0.02	0.00	1.27	0.02	0.30	0.00	0.02	0.00	1.64	0.03	0.03
201001	0.05	0.39	0.18	16.18	0.09	0.70	0.00	0.02	0.00	1.25	0.02	0.30	0.00	0.02	0.00	1.60	0.03	0.03

Tables 1 and 2 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

Table 2: Performance of the models on `SNDlib` instances

	B						W						S					
name	time		nodes		fail	rate	time		nodes		fail	rate	time		nodes		fail	rate
	avg	max	avg	max			avg	max	avg	max			avg	max	avg	max		
abilene	0.00	0.01	0.00	0.08	0.23	0.78	0.00	0.01	0.00	0.00	0.07	0.46	0.00	0.01	0.00	0.00	0.06	0.16
atlanta	0.00	0.01	0.01	0.49	0.18	0.82	0.00	0.01	0.00	0.03	0.05	0.34	0.00	0.01	0.00	0.04	0.04	0.07
cost266	0.00	0.03	0.01	0.82	0.10	0.80	0.00	0.01	0.00	0.03	0.02	0.39	0.00	0.01	0.00	0.01	0.02	0.03
dfn-bwin	0.00	0.01	0.00	0.00	0.17	0.92	0.00	0.01	0.00	0.00	0.00	0.81	0.00	0.01	0.00	0.00	0.00	0.00
dfn-gwin	0.00	0.01	0.00	0.00	0.21	0.91	0.00	0.01	0.00	0.00	0.02	0.66	0.00	0.01	0.00	0.00	0.02	0.02
di-yuan	0.01	0.06	0.53	15.46	0.33	0.84	0.00	0.01	0.00	0.22	0.04	0.41	0.00	0.01	0.01	0.20	0.17	0.18
france	0.00	0.02	0.01	0.40	0.19	0.84	0.00	0.01	0.00	0.00	0.03	0.35	0.00	0.01	0.00	0.01	0.02	0.04
geant	0.00	0.02	0.02	0.67	0.18	0.84	0.00	0.01	0.00	0.09	0.05	0.41	0.00	0.01	0.00	0.06	0.05	0.07
germany50	0.01	0.09	0.07	4.60	0.16	0.84	0.00	0.01	0.00	0.23	0.04	0.37	0.00	0.01	0.00	0.02	0.04	0.06
giul39	0.07	0.78	0.66	36.50	0.24	0.87	0.01	0.03	0.00	1.32	0.00	0.15	0.01	0.04	0.00	0.93	0.00	0.01
india35	0.01	0.06	0.02	1.33	0.19	0.85	0.00	0.01	0.00	0.31	0.04	0.35	0.00	0.01	0.00	0.02	0.04	0.07
janos-us	0.04	0.31	0.29	11.56	0.17	0.81	0.00	0.03	0.03	2.96	0.01	0.17	0.00	0.03	0.01	1.83	0.02	0.02
janos-us-ca	0.07	0.59	0.66	17.72	0.19	0.83	0.00	0.04	0.02	3.28	0.01	0.17	0.01	0.05	0.01	1.99	0.01	0.02
newyork	0.00	0.02	0.02	0.96	0.23	0.88	0.00	0.01	0.00	0.20	0.08	0.42	0.00	0.01	0.00	0.05	0.08	0.12
nobel-eu	0.00	0.04	0.04	1.85	0.18	0.83	0.00	0.01	0.01	0.28	0.06	0.39	0.00	0.01	0.00	0.00	0.07	0.10
nobel-ger	0.00	0.02	0.01	0.45	0.20	0.80	0.00	0.01	0.00	0.00	0.05	0.45	0.00	0.01	0.00	0.00	0.04	0.07
nobel-us	0.00	0.01	0.00	0.00	0.08	0.75	0.00	0.01	0.00	0.02	0.00	0.47	0.00	0.01	0.00	0.01	0.00	0.01
norway	0.01	0.10	0.12	4.24	0.27	0.74	0.00	0.02	0.02	1.22	0.05	0.22	0.00	0.02	0.01	0.64	0.06	0.08
pdh	0.01	0.04	0.28	4.06	0.26	0.74	0.00	0.01	0.05	1.78	0.08	0.28	0.00	0.01	0.01	0.37	0.11	0.11

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

	B						W						S					
name	time		nodes		fail	rate	time		nodes		fail	rate	time		nodes		fail	rate
	avg	max	avg	max			avg	max	avg	max			avg	max	avg	max		
w1-100	0.27	4.85	2.16	35.35	0.28	0.78	0.01	0.04	0.00	0.37	0.00	0.11	0.01	0.07	0.00	0.17	0.00	0.00
w1-200	4.66	95.66	14.27	224.06	0.23	0.79	0.02	0.07	0.00	0.07	0.00	0.12	0.04	0.45	0.01	4.25	0.00	0.00

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, due to a very limited number of cases in which W models experience numerical issues. Even in the few cases where this is visible (di-yuan and pdh), the number of flows that fail due to these problems is small, the large difference in ratio being due to the fact that the overall number of failures is small as well. In fact, conversely, S always allocates visibly higher rates than W, as the theory dictates. Hence, solving ADCSP using W models should, in real-world usage, translate into 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 [30], 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.

All the above observations are confirmed in the much larger `Waxman` topologies (Table 3). Here running times are remarkably and unexpectedly different: the mean running time of B models is over two orders of magnitude larger than that of S or W ones, and the maximum is well over one order of magnitude larger. 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, unlike what was found in [1], the average running time is compatible with real-time usage. This is promising, especially considering that no attempt has been done to reduce running times: the solver was run with all default parameters, and single-threaded.

A more detailed recount of the results is given in Table 4 for three specific instances (each taken from a different set). In the Table, the same statistics are reported, this time with the details of each scheduler class for the same class of models.

Table 4: Performance of the models and schedulers on individual instances

		w1-200						atlanta						199904					
model		time		nodes		fail	rate	time		nodes		fail	rate	time		nodes		fail	rate
		avg	max	avg	max			avg	max	avg	max			avg	max	avg	max		
B	SRP	3.65	110.99	12.02	161.99	0.00	0.80	0.00	0.01	0.00	0.01	0.05	0.85	0.01	0.03	0.00	0.00	0.12	0.98
	GSRP	4.46	83.81	7.84	233.32	0.84	1.00	0.00	0.01	0.00	0.00	0.97	1.00	0.01	0.03	0.00	0.00	1.00	1.00
	WRP	6.06	110.28	18.86	262.87	0.00	0.64	0.00	0.02	0.02	0.90	0.04	0.73	0.01	0.03	0.00	0.14	0.12	0.55
	FB	4.26	65.72	11.94	247.31	0.69	0.93	0.00	0.01	0.01	0.55	0.43	0.89	0.01	0.07	0.03	1.60	0.17	0.66
S	SRP	0.02	0.07	0.00	0.08	0.00	0.12	0.00	0.01	0.00	0.04	0.04	0.33	0.00	0.01	0.00	0.11	0.12	0.52
	WRP	0.02	0.07	0.00	0.08	0.00	0.12	0.00	0.01	0.00	0.04	0.04	0.33	0.00	0.01	0.00	0.11	0.12	0.52
	FB	0.02	0.08	0.00	0.06	0.00	0.12	0.00	0.01	0.00	0.01	0.07	0.35	0.00	0.02	0.00	0.20	0.12	0.52
W	SRP	0.04	0.44	0.01	4.35	0.00	0.00	0.00	0.01	0.00	0.05	0.03	0.06	0.00	0.02	0.00	0.12	0.12	0.12
	WRP	0.04	0.45	0.01	4.35	0.00	0.00	0.00	0.01	0.00	0.05	0.03	0.06	0.00	0.02	0.00	0.12	0.12	0.12
	FB	0.04	0.47	0.01	4.07	0.00	0.00	0.00	0.01	0.00	0.04	0.06	0.08	0.00	0.02	0.00	0.10	0.12	0.12

The Table shows that GSRP (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 GSRP's poor performance were due to a pessimistic upper-bound approximation. Despite this, GSRP 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 typically the second-worst after GSRP, in terms of both failures and reserved rates. This is also expected, due to its latency formula having one more term than WRP's, which in turn has one more term than SRP's. 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 rate 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 never worse than FB, and it can be better than SRP for the same reason. Indeed, this phenomenon shows up in both the `w1-200` and the `atlanta` instances. Remarkably, this does not reflect the actual behavior of the scheduler, but it is an artifact due to the bound assumption (5). In fact, this only happens in B models: for S and W ones the artifact disappears, and the natural ordering is always observed where SRP allocates the least, followed by WRP and 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. 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. This is only visible in the `w1-200` topology for the W model, which have longer running times, but even there the difference is negligible.

In summary, 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 one to do away with the bound assumption (5) that has invariably been employed so far. We believe this paves the way to interesting research developments, as discussed in the next section.

7. Conclusions and future research

In this paper, we have extended the set of 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 GSRP, and the network performance. Characterizing this trade-off will, however, requires 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 something that would seldom (if ever) happen during a simulation. Some initial results in this direction have already been obtained [30], but only limited to a subset of the models. In particular, the results there seem to confirm that the choice of whether or not to employ the bound assumption (5) (i.e., between models B, S and W, where allowed—which is not for GSRP) may be far more relevant than the choice of the scheduler. However, a nontrivial work is still required to confirm this trend experimentally.

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 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 [30]. Any of these possible improvements is very likely to result in much more challenging models due to either a much larger size, or the fact that our 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.

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. All the authors gratefully acknowledge the contribution of the three anonymous referees and of the associate editor of the journal in significantly improving the presentation of this work.

References

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

[2] T. Korkmaz, M. Krunz, Multi-constrained optimal path selection, in: Proceedings of INFOCOM'01, 2001, pp. 834–843.

[3] X. Yuan, Heuristic algorithms for multiconstrained QoS routing, ACM/IEEE Trans. on Networking 10 (2) (2002) 244–256.

[4] A. Juttner, B. Szviatovszki, I. Mecs, Z. Rajko, Lagrange relaxation based method for the QoS routing problem, in: Proceedings of INFOCOM'01, 2001, pp. 859–868.

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

[6] J. Bennett, H. Zhang, WF2Q: Worst-case fair weighted fair queueing, in: INFOCOM, 1996, pp. 120–128.

[7] Q. Ma, P. Steenkiste, Quality-of-Service Routing for Traffic with Performance Guarantees, in: In Proc. IFIP International Workshop on Quality of Service, 1997, pp. 115–126.

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

[9] A. Frangioni, L. Galli, G. Stea, Optimal Joint Path Computation and Rate Allocation for Real-time Traffic, The Computer Journal 58 (6) (2015) 1416–1430.

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

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

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

[13] M. Shreedhar, G. Varghese, Efficient fair queueing using deficit round robin, IEEE/ACM Transactions on Networking 4 (1996) 375–385.

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

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

[16] A. Lori, G. Stea, G. Vaglini, Towards resource-optimal routing plans for real-time traffic, in: T. Margaria, B. Steffen (Eds.), Leveraging Applications of Formal Methods, Verification, and Validation, Vol. 6415 of Lecture Notes in Computer Science, Springer-Verlag, 2010, pp. 214–227.

[17] J.-Y. L. Boudec, P. Thiran, Network Calculus: A Theory of Deterministic Queuing Systems for the Internet, Springer LNCS, 2001.

[18] J. Xu, R. Lipton, On fundamental tradeoffs between delay bounds and computational complexity in packet scheduling algorithms, in: SIGCOMM, 2002, pp. 279–292.

[19] P. Valente, Exact gps simulation with logarithmic complexity, and its application to an optimally fair scheduler, in: SIGCOMM, 2004, pp. 269–280.

[20] P. Valente, Private communication (2014).

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

[22] A. Frangioni, L. Galli, G. Stea, Delay-constrained routing problems: Accurate scheduling models and admission control, Technical Report, Dipartimento di Informatica, Università di Pisa (2015).

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

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

[25] Garr, `http://www.garr.it`.

[26] The internet topology zoo, `http://www.topology-zoo.org`.

[27] Sndlib, `http://www.sndlib.zib.de`.

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

[29] L. Saino, C. Cocora, 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, 2013, pp. 82–91.

[30] A. Frangioni, L. Galli, G. Stea, QoS routing with worst-case delay constraints: Models, algorithms and performance analysis, Computer Communications to appear.