Optimizing revenue for bandwidth auctions over networks with time reservations

Pablo Belzarena[*,a], Fernando Paganini[b], Andrés Ferragut[b]

[a]*Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay*
[b]*Facultad de Ingeniería, Universidad ORT, Montevideo, Uruguay*

Abstract

This paper concerns the problem of allocating network capacity through periodic auctions, in which users submit bids for fixed amounts of end-to-end service. We seek a distributed allocation policy over a general network topology that optimizes revenue for the operator, under the provision that resources allocated in a given auction are reserved for the entire duration of the connection.

We first study periodic auctions under reservations for a single resource, modeling the optimal revenue problem as a Markov Decision Process (MDP), and developing a receding horizon approximation to its solution. Next, we consider the distributed allocation of a single auction over a general network, writing it as an integer program and studying its convex relaxation; techniques of proximal optimization are applied to obtain a convergent algorithm. Combining the two approaches we formulate a receding horizon optimization of revenue over a general network topology, leading to a convex program with a distributed solution. The solution is also generalized to the multipath case, where many routes are available for each end-to-end service. A simulation framework is implemented to illustrate the performance of the proposal, and representative examples are shown.

Key words: bandwidth auctions, Markov decision processes, utility maximization

1. Introduction

The possibility of auctioning bandwidth in real time has been considered by many authors [15, 11, 18, 23, 9, 24], with a variety of applications: diffserv, access control, 3G cellular access, VPNs, etc. Much of this work has focused on game-theoretic considerations, in particular on providing incentives for bidders to reveal their true utilities. The standard theory of auctions [14] provides these mechanisms for the auctioning of a single resource, but it is far more challenging to extend them to a general network topology. Most proposals in this regard require the user (or a broker entity acting on his/her behalf), to place separate bids for internal resources of the network. In particular, the Progressive Second Price (PSP) mechanism of [15] requires each player to coordinate bids at the different nodes on its route, so that each node may run an auction with the allocation and pricing rules of the single resource case. PSP has a long convergence phase, which is improved by a *multibid* method in [18]; however, the latter mechanism only applies to tree topologies. Another approach to bandwidth auctioning for multicast trees or VPNs is proposed in [9], based on Dutch auctions. The mechanism assumes that users interested in a path would try to reserve bandwidth by placing bids simultaneously for all constituent links.

In this paper we argue that to have practical impact, a bandwidth auction requires a simpler user interface: the consumer should submit a bid for an entire end-to-end service, oblivious of the internal topology. It is the operator's problem to decide which of these bids to accept and how to accommodate the aggregate service within the available network capacity. Furthermore, a more natural objective than incentive compatibility is revenue maximization for the operator that offers this end-to-end service. As one possible deployment scenario to make the discussion concrete, consider the Service Overlay Network (SON) architecture [12], where an overlay operator has leased tunnels between a set of service gateways located in domain boundaries, and auctions a service of high-quality (e.g. video-on-demand) over this infrastructure, with the objective of obtaining revenue.

Another important aspect of the problem that has not been satisfactorily addressed in previous work are

[*]Corresponding author
Email addresses: belza@fing.edu.uy (Pablo Belzarena), paganini@ort.edu.uy (Fernando Paganini), ferragut@ort.edu.uy (Andrés Ferragut)

inter-temporal considerations. Most references cover a one-shot auction where bids for the entire duration are known initially. References for multi-period auctions (e.g. [23]) allow future bidders to compete with incumbent ones, albeit given the latter some advantage. This is not an attractive condition for our intended applications. Consider for example selling video-on-demand content about 100 minutes long, in auctions every 5 minutes. A consumer will not purchase the service if he/she faces the risk of losing the connection close to the end of the movie. In this paper we impose the condition that once bandwidth has been allocated in an auction, the successful bidder has a *reservation* for the duration of his/her connection. This means that the operator must assume the risk of future auctions, which makes the maximization of revenue a stochastic dynamic optimization problem.

Both of the above aspects (general network topology, time reservations) lead to optimization problems of high complexity, on top of which we add the requirement of a distributed solution. Rather than an exact solution, we develop in this paper a series of tractable methods that approximate the optimal revenue objective. We begin in Section 2 with auctions of a single resource (single link capacity) with time-reservations, a problem that we formulate as a Markov decision process (MDP) [1, 21]. We introduce a receding horizon approximation that is able to capture the dynamic component of the problem in a tractable way, and validate it by simulation. Next, we turn in Section 3 to the network aspect, formulating the allocation of a one-shot auction as an integer program; by recasting this problem in the language of Network Utility Maximization (NUM) [13, 7], we develop a natural relaxation that has a distributed solution; convergence is obtained through the application of a proximal optimization method [5, 16].

In Section 4 we combine the previous approaches to formulate a receding-horizon optimization of revenue for multi-period auctions over a distributed network, which again is formulated as a variant of a NUM problem, solved in relaxed form through a proximal method. We develop in this case a distributed implementation of the algorithm, and exhibit its performance in a series of simulation examples that progressively include more realistic situations. Finally, in Section 5 we consider multipath optimization, where end-to-end services can be offered through multiple routes inside the network; we show how to extend the methodology to this case. Conclusions are given in Section 6.

This article is an extension of our conference papers [2, 3]. One main enhancement included here is the proximal approximation method to ensure convergence of our distributed algorithms with non-strictly concave utilities. Also, the entirety of Section 5 on multipath auctions is new material.

2. Periodic auctions of a single resource with time reservations

We consider first an auction for the capacity of a single resource, the bandwidth of one link, postponing the consideration of network topology. The focus here is the temporal dimension: auctions are held periodically, based on bids collected for a period of time of duration T. When each auction closes, the provider decides which bidders are allocated capacity, which is subsequently *reserved* for a service duration that may exceed T. In particular, when the next auction occurs, new bidders are not allowed to displace incumbent users. The objective is to find an allocation policy that maximizes revenue of the seller over time, under the assumption that users pay their bid upon admittance to the service, a first-price auction. Later on we discuss strategic implications.

We establish some notation. Let σ be the bandwidth requirement of the single service being auctioned; the provider has capacity c to auction. In this section we normalize $\sigma = 1$, and assume c is an integer.

The discrete time index k defines the auction at time kT, for which the seller has received N^k bids, ordered as

$$b^{k,(1)} \geq b^{k,(2)} \geq \cdots \geq b^{k,(N^k)}.$$

The result of the auction is a capacity allocation a^k to a set of highest bidders, yielding a revenue of

$$U_{b^k}(a^k) := \sum_{i=1}^{a^k} b^{k,(i)}. \quad (1)$$

This function $U_{b^k}(\cdot)$ is defined above for integer values of a^k; we will also apply this notation to the function of $a^k \in \mathbb{R}$ defined by linear interpolation, and constant above N^k. This piecewise linear function is increasing and concave in a^k, since bids are decreasing.

If we were considering a single auction of the capacity c, clearly the optimal revenue decision would be to sell as much as possible, $a^k = \min\{c, N^k\}$. However, the occurrence of periodic auctions and reservations across multiple periods complicates the decision significantly, as discussed next.

2.1. Optimal allocation as a Markov Decision Process

The long-term optimal revenue problem is posed in terms of a stochastic model for the bidding and duration processes. The model assumptions are now described:

- **Distribution of bids.** We assume bids are drawn independently from a continuous probability distribution. For the theory to follow, we will assume the distribution is known; in Section 4.2 we show how it can be learned from past observations.

- **Number of bids.** Two alternatives are considered:

  - A fixed number $N^k = N$ of bids;
  - A random number of bids, with Poisson distribution of parameter λT, that results from a Poisson process of bid arrivals with rate λ.

- **Revenue functions**. First-price charging is assumed. The revenue function for given bids is (1), and we define the expected revenue function as

$$\overline{U}(a) = E[U_b(a)]. \tag{2}$$

  Here the expectation is over the bid distribution and possibly the number of bids. As is the case with (1), this function is also piecewise linear and concave in a, at integer values representing the expected revenue from admitting a connections.

- **Service duration.** As explained before, connections are reserved for the entire duration, which is characteristic of the service being auctioned. To allow for a Markovian analysis, we will use a stochastic, memoryless model: service durations are independent exponential random variables, of mean $1/\mu$. Therefore at the end of the period T each connection has probability $p := e^{-\mu T}$ of remaining active for the following period.

We now describe the process dynamics . Let x^k denote the number of connections active at $t = kT^-$, i.e. before the k-th auction. The system admits a^k new connections, $0 \le a^k \le c - x^k$, taking the total to $x^k + a^k$. By the next auction period, $t = (k+1)T^-$, the number of active connections x^{k+1} follows then a binomial distribution with parameters $x^k + a^k$ and p:

$$P[x^{k+1} = i | x^k, a^k] = \binom{x^k + a^k}{i} p^i (1-p)^{x^k + a^k - i}. \tag{3}$$

We our now ready to formulate our first stochastic optimization problem.

Problem 1 (Optimal mean revenue, single link).

$$\textit{Maximize} \quad \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} E[U_{b^k}(a^k)].$$

Here the expectation is over two sources of randomness: the vector of bids b^k and the departure process. The constraints are $0 \le a^k \le c - x^k$ where x^k follows the binomial transition dynamics (3). We can also consider the discounted version:

$$\text{Maximize} \quad \sum_{k=0}^{\infty} \rho^k E[U_{b^k}(a^k)], \quad \text{where } 0 < \rho < 1.$$

Both are Markov Decision Processes (MDPs) [4, 21]. The *state* at time k is given by (x^k, b^k), i.e. the current occupation and the incoming bids. Based on this state, the *action* $a^k = a(x^k, b^k)$ decides on how many bids to accept. A solution to the MDP is a *policy* $a(x, b)$ that results in a minimum cost. In the discounted case $\rho < 1$, this policy satisfies the Bellman equation

$$V^*(x^0, b) = \max_{a\in\mathcal{A}_x} \left\{ U_b(a) + \rho E[V^*(x^1, b')] \right\}, \tag{4}$$

where V^* is the value function and the expectation is taken over the binomial distribution of $x^1 | (x^0, a)$ and the distribution of the next bid b'. The state-dependent constraints are $\mathcal{A}_x = \{0 \le a \le c - x^0\}$. For $\rho = 1$, V^* satisfying (4) is no longer the optimal cost, but (4) still characterizes the optimal action $a(x, b)$.

It is in general difficult to solve the Bellman equation; a commonly used strategy is the *value iteration* [4]

$$V_{m+1}(x^0, b) := \max_{a\in\mathcal{A}_x} \left\{ U_b(a) + \rho E[V_m(x^1, b')] \right\};$$

starting with an arbitrary $V_0(x, b)$, $V_m(x, b)$ converges to $V^*(x, b)$, and the corresponding maximizing action converges to the optimal action [4].

2.2. Receding horizon approximation.

We use initial steps of the value iteration to approximate the optimal policy. Starting from $V_0 \equiv 0$, we have

$$V_1(x^0, b) = \max_{a\le c-x^0} U_b(a) = U_b(c - x^0).$$

This first step gives the "myopic" policy $a = c - x^0$, that sells all available capacity without regard to the future. This is clearly suboptimal, but may be appropriate for certain parametric scenarios. To improve on it, we take a second step in the value iteration:

$$\begin{aligned} V_2(x^0, b) &= \max_{a\le c-x^0} \{ U_b(a) + \rho E[V_1(x^1, b')] \} \\ &= \max_{a\le c-x^0} \{ U_b(a) + \rho E[U_{b'}(c - x^1)] \} \\ &= \max_{a\le c-x^0} \{ U_b(a) + \rho E_{x^1} \overline{U}(c - x^1) \}. \end{aligned}$$

In the last step we have taken expectation with respect to the bid b', using $\overline{U}$ defined above; what remains is the expectation with respect to $x^1 \sim \mathrm{Bin}(x^0 + a, p)$. The above optimization can be given a *receding horizon* interpretation: optimize over the current revenue plus the expected revenue of looking one step ahead, assuming all available capacity will be sold off at that time. This decision is applied recursively; thus the future is taken into account, but at a limited level of complexity.

The receding horizon policy is thus the following: at each auction k, let x^k denote the current occupation, and b^k the vector of incoming bids. Admit the number of bids a^k that solves

Problem 2 (Receding horizon policy, single link).

$$\max_{a \le c - x^k} \{U_b(a) + \rho E_X \overline{U}(c - X)\}, \tag{5}$$

where the expectation is over $X \sim Bin(x^k + a, p)$.

We now analyze how to carry out this optimization. The first term in (5) increases with a. To characterize the second, we rewrite it as follows. Consider the function $W(i) = \overline{U}(c) - \overline{U}(c - i)$, piecewise linear, increasing and *convex* in i. Indeed, the increments

$$w(i) := W(i+1) - W(i) = E[b^{(c-i)}], \quad i = 1, \dots, c$$

are non-negative and increasing in i (since bids are decreasing). We now study the expectation with respect to the binomial distribution.

Proposition 1. *Define $\overline{W}(x) = E[W(I_x)]$, where $I_x \sim Bin(x, p)$ for integer x, and extend by linear interpolation. Then $\overline{W}(x)$ is increasing and convex.*

Proof 1. *Given $I_x \sim Bin(x, p)$, integer x, we can generate a $Bin(x+1, p)$ random variable of the form $I_x + \xi$, where ξ is Bernoulli(p), independent of I_x. Writing*

$$W(I_x + \xi) - W(I_x) = w(I_x + 1)\xi$$

and taking expectations, using independence we obtain increments

$$\overline{w}(x+1) := \overline{W}(x+1) - \overline{W}(x) = pE[w(I_x + 1)]. \tag{6}$$

It remains to show the last term is increasing in x. Noting that $w(i)$ is increasing, the inequality $w(I_x + \xi + 1) \ge w(I_x + 1)$ holds almost surely; taking expectations,

$$E[w(I_{(x+1)} + 1)] = E[w(I_x + \xi + 1)] \ge E[w(I_x + 1)].$$

The receding horizon optimization (5) can now be rewritten as

$$\max_{a \le c - x^0} U_b(a) - \rho \overline{W}(x^0 + a) + \rho \overline{U}(c). \tag{7}$$

Implicit in (5) and (7) is that a is an integer. In this case, however, the condition can be relaxed without loss of generality, treating (7) as a convex optimization problem. To solve it amounts to looking for a crossing point between the derivatives of $U_b(a)$ and $\rho \overline{W}(x^0 + a)$ (marginal utilities and costs), as depicted in Fig. 1.

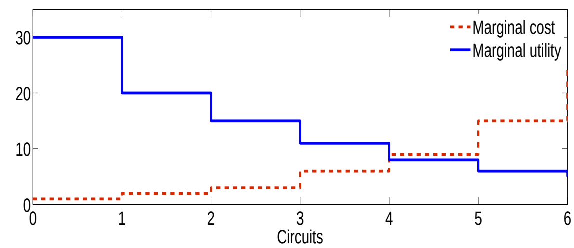

Figure 1: Marginal utility versus marginal cost

The marginal utilities are just the current bids in decreasing order. The marginal costs represent the value of leaving one more free circuit for the next auction, and have the form $\rho \overline{w}(i)$, with $\overline{w}(\cdot)$ defined in (6); as noted they are increasing in i. Since the bids b are random, the curves of Figure 1 will almost surely cross at a single, integer point. So the convex relaxation is innocuous.

The optimal acceptance policy is the value a such that

$$b^{(1)} \ge \cdots \ge b^{(a)} \ge \rho \overline{w}(i) > b^{(a+1)}, \quad \text{for } i = x^0 + a.$$

The values $\rho \overline{w}(i)$; act as successive *thresholds*: to accept a bids, the *lowest* one must exceed $\rho \overline{w}(x^0 + a)$. To accept one more, we require a *more demanding* threshold $\rho \overline{w}(x^0 + a + 1)$ on this (smaller) bid.

A concrete formula for the thresholds as a function of the bid distribution is given (see the Appendix) by

$$\overline{w}(i) = p \sum_{l=0}^{i-1} E(b^{(c-l)}) \binom{i-1}{l} p^l (1-p)^{i-1-l}. \tag{8}$$

Based on knowledge of ρ, p, and the distribution of bids, this expression could be calculated offline and used for carrying out auctions with the policy (5).

Example 1. *We evaluate the previous results in a few simple cases (for $\rho = 1$). For $c = 1$, there is a single link cost $\overline{w}_1 = pE(b^{(1)})$, that acts as an admission threshold for bids received when the circuit becomes empty. For* *instance in the case of N bids, uniformly distributed in* $[0, b_{max}]$ *we have* $\overline{w}_1 = p\frac{N}{N+1}b_{max}$.

If $c = 2$*, there are two marginal costs:* $\overline{w}_1 = pE(b^{(2)})$ *for occupying the first connection, and* $\overline{w}_2 = p[E(b^{(2)})(1-p) + E(b^{(1)})p]$ *for occupying the second. For uniform in* $[0, b_{max}]$ *bids we have*

$$\overline{w}_1 = p\,\frac{N-1}{N+1}b_{max}$$
$$\overline{w}_2 = p\,\frac{(N-1)(1-p) + Np}{N+1}\,b_{max}.$$

We now compare by simulations our receding horizon policy with the optimal infinite-horizon MDP, in the case of one circuit ($c = 1$). In this simple case, the latter is also a threshold policy on the bids, but the optimal threshold does not have a simple formula; we computed it numerically through the value iteration algorithm from [10]. Fig. 2 shows the acceptance thresholds for both policies: we see the infinite horizon threshold is more demanding. Fig. 3 shows the average utility obtained by simulation of these two policies. Results are very similar. Therefore, in this case we have managed to extract almost the optimal utility just by looking one-step ahead with the policy. On the other hand, if we apply the myopic policy that always fills the link, the second plot shows there is a clear loss in utility.

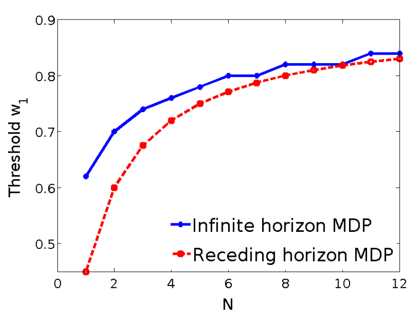

Figure 2: Threshold comparison between policies, $c = 1$, $p = 0.1$.

2.3. Strategic and game considerations

A large focus of the auction literature has been strategic bidding, and the design of mechanisms in which bidding true utilities is a dominant strategy. Vickrey's second-price auction [26], where the winning user is charged the second highest bid, is of this kind. More generally, VCG mechanisms (for Vickrey-Clarke-Groves, see e.g. [8]) have built-in "incentive compatibility", a condition sought in many auction designs for networks [15, 18, 9]. In contrast, we have proposed a first-price auction with incoming bids, which does not have built-in incentives for revealing true utilities; in this section we explain the reasons behind our choice.

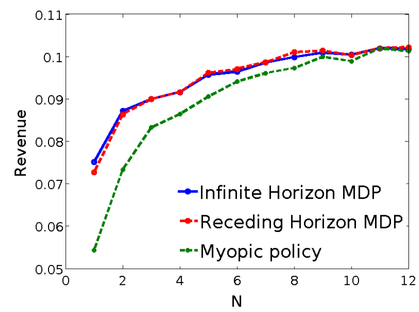

Figure 3: Revenue comparison between policies, $c = 1$, $p = 0.1$.

The main reason is that our stated objective is revenue maximization, rather than truth-revelation. A fundamental result of the theory of auctions, the *Revenue Equivalence Theorem*, see [8], states that under certain assumptions (mainly, risk neutrality of participants) all auctions have the same expected revenue for the seller. However, under other conditions (e.g., risk-averse buyers) first-price auctions are known to improve revenue [8]. To illustrate this issue in a simple setting, consider a one-shot auction of capacity c, avoiding the temporal dimension. If in a certain auction there are fewer than c competitors, the generalized Vickrey auction would charge a price equal to the highest bid left out, in this case, zero, hence the network receives zero revenue. If, instead, we charge users what they bid, how would strategic bidders behave? *If they knew* that capacity is not scarce, the rational thing would be to submit a bid close to zero; this would confirm revenue equivalence. However, in a practical situation they would not have this information, and will be compelled to bid a non-negligible amount. So the seller is better off with a first-price auction.

A second consideration is the *complexity* of truth revealing mechanisms, when we add the time dimension, and later on the network topology. In this regard, we note:

- For a generalized Vickrey auction, the revenue function (1) is no longer concave, or increasing; for instance the vector of bids $b = (2, 2, 1, 1)$ gives revenue values $(2, 2, 3, 0)$. These properties are essential to the tractability of our optimization over time.
- In the network case, to be discussed in future sec-

tions, the generalization of VCG mechanisms is very complex, even for a one-shot auction. This issue was already brought up in [19]. Namely, finding the VCG charges involves solving multiple optimization problems, with each bidder removed from the network.

3. Distributed one-shot auctions over a network

We turn now to auctioning bandwidth over a general network topology. We seek methods to optimize revenue of the allocated bids, that can be computed in a distributed way across a network. This section focuses on a single auction of available capacity; later on we will return to the temporal considerations associated with reservations.

We begin by extending our notation to the network case. The network is composed of a set of links indexed by l, and a set of end-to-end routes indexed by r. R denotes the routing matrix, $R_{lr} = 1$ iff route r includes link l, otherwise $R_{lr} = 0$. $c = (c_l)$ is the vector of link capacities.

Consider a set of services, indexed by s, that are auctioned over the network. Each service has a fixed bandwidth requirement σ_s: users bid for this well-defined rate allocation. For the moment we consider the *single path* situation, in which service s is offered over a single route $r(s)$; in Section 5 we extend the method to the multipath setting. There could be multiple services offered over the same route. For each s, the network receives a set of N_s bids $b_s^{(i)}$, ordered as

$$b_s^{(1)} \geq b_s^{(2)} \geq \cdots \geq b_s^{(N_s)}.$$

The resource allocation decision is to find which of these bids to accept, within the capacity constraints of the network, to maximize revenue under a first-price auction. Defining the variable $\xi_{s,i}$ by $\xi_{s,i} = 1$ if bid $b_s^{(i)}$ is accepted, $\xi_{s,i} = 0$ otherwise, the optimal revenue problem for the single auction is the integer program

$$\max \sum_s \sum_{i=1}^{N_s} b_s^{(i)} \xi_{s,i} \tag{9a}$$

$$\text{subject to} \quad \sum_s \sum_{i=1}^{N_s} R_{lr(s)} \sigma_s \xi_{s,i} \leq c_l \quad \forall l, \tag{9b}$$

$$\xi_{s,i} \in \{0, 1\}. \tag{9c}$$

We can also convert this problem to a utility-based form as in the previous section. Since for fixed s, all bids $b_s^{(i)}$ are for the same amount of bandwidth, the optimal solution will involve the highest bids per service,

$$\sum_{i=1}^{N_s} b_s^{(i)} \xi_{s,i} = \sum_{i=1}^{m_s} b_s^{(i)},$$

where the integer variable m_s is the number of accepted bids for service s, resulting in a rate $a_s := \sigma_s m_s$ for this service applied to route $r(s)$. Now define

$$U_{b_s}(a_s) := \sum_{i=1}^{a_s/\sigma_s} b_s^{(i)}. \tag{10}$$

This function is defined above for discrete values of a_s (the multiples of σ_s). As before we extend it to a piecewise linear, concave function of $a_s \in \mathbb{R}$, by linear interpolation. With this notation, we rewrite (9) as follows.

Problem 3 (Network allocation, single auction).

$$\max \sum_s U_{b_s}(a_s) \tag{11a}$$

$$\text{subject to} \quad \sum_s R_{lr(s)} a_s \leq c_l \quad \forall l, \tag{11b}$$

$$a_s/\sigma_s \in \mathbb{Z}. \tag{11c}$$

3.1. Convex relaxation and distributed solution

If we ignore the integer constraint in (11c), the optimization in Problem 3 has the form of the *network utility maximization* problem in congestion control [13, 17, 25], that is known to have distributed solutions. Motivated by this, we will focus on the convex relaxation (11a-11b). Before doing this, we inquire whether it can be claimed that the relaxation is generically exact (equivalently, whether the linear program that relaxes (9) has generically integer solutions). Unfortunately, this is not the case.

Example 2. *Consider 4 links with capacity $c_l = 2$, and 5 paths (each with bandwidth requirement $\sigma_s = 1$), with routing matrix*

$$R = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Bids for the same route are all equal, with the following distribution among routes: $b_1 = b_2 = b_3 = b_4 = 1$, and $1 < b_5 < \frac{4}{3}$. Then, the relaxed convex program (11a-11b) has solution $a^* = (\frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, 0)^T$, with optimum revenue $U^* = \frac{8}{3}$. To see this, note first that a^* satisfies (11b) with equality. Now consider the prices

$\alpha_l^* = \frac{1}{3}$, $l = 1,2,3,4$, with aggregate route prices $q^* = (1,1,1,1,\frac{4}{3})^T$. Since $b_5 < q_5^*$, we must have $a_5 = 0$, but the remaining coordinates are indeterminate in $[0,1]$. So the proposed point (a^*, α^*) is a saddle, but this would not happen with integer coordinates in a.

In fact, here the integer program can be solved by observing that at most two connections can be active over all routes, so the best solution is to give them to the highest bidders in route five, $\tilde{a} = (0,0,0,0,2)^T$. This gives an optimal integer revenue $\tilde{U} = 2b_5 < \frac{8}{3}$. So the optimal relaxed solution is better than any integer solution.

The above example shows that optimal revenue is not an easy integer program, its convex relaxation is not exact. Since integer programming is NP hard, we have strong indication of a fundamental difficulty in this problem, not easy to overcome even allowing for centralized computation. We will thus accept a sub-optimal allocation: solving the convex relaxation, and rounding off to satisfy the integer constraints. Note from the above example that this may not be optimal; if, however, the capacity allows for a large number of connections, the loss of revenue is moderate. In compensation, we will obtain an efficient distributed solution.

Remark 1. *Our problem shares similarities with an optimal resource allocation problem studied in [20], under opposite conditions: fixed input demand, minimization of a convex cost subject to integer constraints. Again, except for special cases this integer program does not have an exact convex relaxation.*

We thus focus on finding a distributed solution to the convex program (11a-11b), for which we can draw from duality methods used in congestion control. The associated Lagrangian is

$$L(a,\alpha) = \sum_s U_{b_s}(a_s) + \sum_l \alpha_l [c_l - \sum_s R_{lr(s)} a_s] = \sum_s [U_{b_s}(a_s) - q_{r(s)} a_s] + \sum_l \alpha_l c_l,$$

where $\alpha = (\alpha_l)$ is a vector of Lagrange multipliers (prices) associated with the constraints (11b), and $q_r = \sum_l R_{rl}\alpha_l$ are the accumulated route prices. A gradient-projection algorithm to find a saddle point of $L(a,\alpha)$ is

$$a_s := \arg\max_{a_s} [U_{b_s}(a_s) - q_{r(s)} a_s], \tag{12a}$$

$$\alpha_l := [\alpha_l + \gamma_l (y_l - c_l)]^+. \tag{12b}$$

Here $y_l = \sum_s R_{lr(s)} a_s$, $[\cdot]^+ = \max\{\cdot, 0\}$ and $\gamma_l > 0$ are step sizes.

(12a) uses current route prices to fix a rate allocation with maximum "surplus" (utility minus a linear cost). (12b) compares the proposed allocation to link capacity and updates prices (up or down) accordingly.

In congestion control, the preceding equations are interpreted as describing the *data plane*, in which elastic sources adapt their packet rate and links generate prices based on their instantaneous congestion. In our situation, we think of the above equations as an iteration in the *control plane*, which is run to settle an auction prior to any allocation of resources. More details on implementation are given in Section 4.

For strictly concave utilities, it is shown in [17] that the above iteration converges to the optimal allocation, for sufficiently small step size. Here, however, U_{b_s} is not strictly concave, it is piecewise linear, changing slope at the multiples of σ_s; this may compromise the convergence of the algorithm. In particular, the optimization (12a) amounts to comparing the marginal utilities $U'_{b_s}(a_s)$ with the current price $q_{r(s)}$. The former are the bids in decreasing order, analogously to the graph in Fig. 1, but now compared with a constant. Here as well the curves will generically cross at an integer multiple of σ_s; however if the relaxed problem has a non-integer optimum, $q_{r(s)}$ will oscillate around the value of one bid, and the resulting a_s will "chatter" between the adjacent integer values.

3.2. Proximal optimization algorithm

One method to obtain convergence is to modify our problem through the so-called *proximal optimization* method [5, 16, 22, 28], described as follows.

Problem 4 (Proximal optimization, single auction).

$$\max \sum_s U_{b_s}(a_s) - \sum_s \frac{\kappa_s}{2}(a_s - d_s)^2 \tag{13a}$$

$$\text{subject to} \quad \sum_s R_{lr(s)} a_s \le c_l \quad \forall l. \tag{13b}$$

In the above, $\kappa_s > 0$ is a constant and d_s an additional free variable, which clearly makes the problem equivalent to the relaxed problem (11a-11b).

The standard Proximal Optimization Algorithm [5] consists of two steps:

(i) For fixed $d = (d_s)$, optimize (13a-13b) over a_s; this is now a strictly concave program.

(ii) For fixed a_s, optimize over d_s, i.e. set $d_s := a_s$.

It is shown in [5] that this kind of iteration converges to the optimum, with a rate of convergence that depends on κ_s and improves as κ_s becomes smaller (see [5] Chapter 3, exercise 4.2). Rockafellar in [22] shows that under mild assumptions the convergence occurs at least at a linear rate, i.e. that the distance to equilibrium decreases by a constant factor per iteration. This factor tends to zero with κ_s, so convergence can be made superlinear by decreasing κ_s as the iteration progresses. In [28], a generalized Newton method is proposed that also solves the standard proximal algorithm with a superlinear rate of convergence.

We note, however, that in our case step (i) above is a constrained problem that must itself be solved through a dual iteration similar to (12); the cited results apply if this dual iteration is assumed to converge before the update (ii) in d_s takes place; this means the algorithm would have have two time-scales. This is unsuitable for online implementation because in practice, it is difficult for the network elements to decide in a distributed fashion when the inner level of iterations should stop.

A more practical alternative for a distributed implementation in a network is to perform a finite number of dual gradient steps per update of d_s, as follows.

(i') For fixed d_s, run N steps of the iteration

$$a_s := \arg\max_{a_s}[U_{b_s}(a_s) - q_{r(s)}a_s - \frac{\kappa_s}{2}(a_s - d_s)^2], \tag{14a}$$

$$\alpha_l := [\alpha_l + \gamma_l(y_l - c_l)]^+. \tag{14b}$$

(ii) Set $d_s := a_s$, and go to (i').

In [16] an algorithm of this kind was studied, for a utility maximization problem that has some similarities with the one considered here. It was proved that provided the step-size in the subgradient step is small enough, the algorithm converges for any value $1 \leq N \leq \infty$. The maximum step-size in order to have convergence depends on κ_s, and decreases with it. Thus the choice of κ_s involves a tradeoff between the rate of convergence of the inner (dual subgradient) iteration and that of the outer proximal algorithm. We will not attempt here to reproduce the theory of [16] for our problem. Rather, in the following sections we will study the behavior of the above iteration by simulation.

As an additional comment, notice that the solution of (14a) reduces to the equation

$$U'_{b_s}(a_s) = q_{r(s)} + \kappa_s(a_s - d_s), \tag{15}$$

which amounts to intersecting the step function of decreasing bids with a straight line, as depicted in Figure 4. The linear term with $\kappa_s > 0$ makes it possible to have an intersecting point a_s which is not an integer multiple of σ_s, as required for solving the relaxed problem.

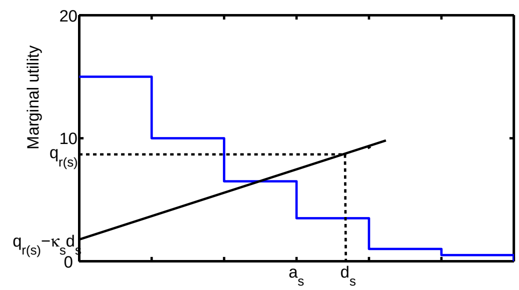

Figure 4: Solving for a_s in the proximal method.

3.3. Simulation Example

In this example we consider the network of the figure 5. First, we auction the capacity of the network using the algorithm defined by equations (12a) and (12b). In figure 6 we show that this algorithm does not converge for Route 1 that oscillates between 4 and 5 connections.

After that, we auction the capacity of the network using the proximal optimization algorithm defined by equations (14a) and (14b). In Figure 7 we show that in this case the algorithm converges for all routes.

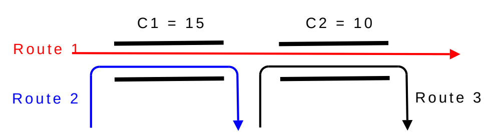

Figure 5: Linear network.

4. Periodic auctions in the network case

Having considered two sub-problems in the previous sections, we now take on the problem of optimizing revenue for periodic bandwidth auctions with time reservations over a general network topology. More specifically: assume that a set of services is defined over a network, each characterized by a route and bandwidth requirement as in Section 3; every time T, a set of bids is collected for each service, and the network must make an allocation decision among all services, within the capacity constraints, that takes into account the future impact of reserved bandwidth.

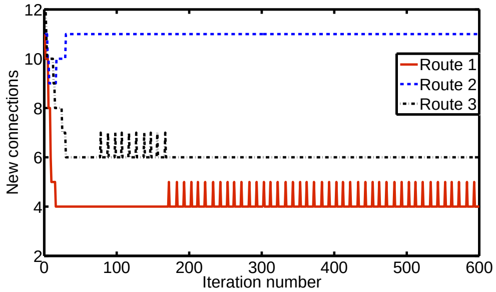

Figure 6: Route 1 does not converge using the algorithm defined by equations (12a) and (12b).

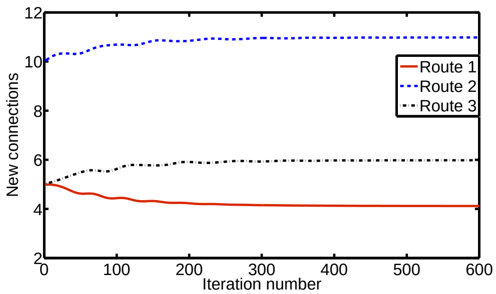

Figure 7: Convergence using the proximal optimization algorithm.

Given the complexity we encountered in each of the two subproblems (the MDP for periodic auctions in the single resource case, an integer program for one-shot auctions in the network case) it should be clear that this unified problem is not tractable in its exact form. We will thus develop an approximation that combines the receding horizon approach of Section 2 with the convex relaxation of Section 3.

4.1. Fluid receding horizon policy

We begin by formulating a variant of the receding horizon policy of Problem 2, more suitable for generalization to the network case. The idea is to replace $E_X\overline{U}(c-X)$ in (5) with $\overline{U}(c-E[X])$, leading to a deterministic fluid optimization, where the expectation of $X \sim \text{Bin}(x^k + a, p)$ can be readily computed to be $p(x^k + a)$. The above change overestimates the one-step ahead utility, since $\overline{U}(\cdot)$ is concave; the error will, however, be moderate as the number of circuits c grows, and the binomial distribution becomes concentrated around its mean. This leads to the following policy for the single resource case: admit the number of bids a^k that solves

$$a^k = \text{argmax}_{a \le c - x^k}\{U_b(a) + \rho\overline{U}(c - p(x^k + a))\}. \quad (16)$$

By introducing a slack variable z we can also rewrite the above optimization as a convex program, as follows:

Problem 5 (Fluid receding horizon policy, single link).

$$\max_{a,z} \; [U_b(a) + \rho\overline{U}(z)],$$
$$\text{subject to} \quad x^k + a \le c,$$
$$p(x^k + a) + z \le c.$$

At the optimum, the constraint in z is an equality, $z = c - p(x^k + a)$, so clearly this problem is equivalent to (16). Note that z (expected future allocation) need not be an integer. On the other hand, a^k should be an integer, something that is *not* guaranteed generically for the optimum of Problem 5; in this sense, Problem 2 is better behaved for the single link case. Problem 5 is, however, more easy to generalize to the network case, where in any event integer constraints will necessarily be relaxed.

The network generalization of the fluid receding horizon policy is now described. In this context, the problem studied in Section 3 covers the myopic policy of auctioning all bandwidth; we wish to incorporate the consideration of future revenue, generalizing the method of Problem 5.

We describe the allocation decision at time $k = 0$, and hence avoid inserting time indices in the bids and other variables. For each service s, we denote by x_s^0 the rate from previous occupation, a_s the rate allocation in the current auction, and z_s the expected rate allocation in the following auction ($t = T$). Recall the definition (10) of the piecewise linear utility $U_{b_s}(a_s)$ based on current bids; analogously define $\overline{U}_s(\cdot)$ as in (2), replacing bids by their expectation. Both are in terms of σ_s, the bandwidth requirement of the class of service associated with s. Another feature of the class of service is the model for duration: let p_s be the probability that a connection active at $t = 0$ will remain active at $t = T$ [1].

Problem 6 (Network receding horizon allocation).

$$\max \sum_s U_{b_s}(a_s) + \rho\overline{U}_s(z_s), \quad (17a)$$

$$\text{subject to} \quad y_l := \sum_s R_{lr(s)}(a_s + x_s^0) \le c_l \quad \forall l, \quad (17b)$$

$$\tilde{y}_l := \sum_s R_{lr(s)}[p_s(a_s + x_s^0) + z_s] \le c_l \quad \forall l. \quad (17c)$$

[1]For an exponential duration, $p_s = e^{-\mu_s T}$.

We have already stated the problem in terms of a convex relaxation, omitting the integer constraint of the form (11c); as before, this constraint is not automatically guaranteed and not easy to enforce, in practice we will round off the solution of Problem 6. Similarly to the situation of Section 3, we also face challenges in obtaining distributed solutions through duality, due to lack of strict concavity of the objective: these can be solved by generalizing the proximal approximation method of Problem 4.

Problem 7. (Proximal network receding horizon allocation)

$$\max \sum_s U_{b_s}(a_s) + \rho\overline{U}_s(z_s) - \frac{\kappa_s}{2}[(a_s - d_s)^2 + (z_s - e_s)^2] \tag{18}$$

subject to (17b-17c).

The above problem includes auxiliary variables d_s, e_s, which at optimality become equal to a_s, z_s respectively, but provide "inertia" when computing a dual solution. We describe the resulting algorithm through the Lagrangian $L(a, z, \alpha, \beta)$, that includes now two vectors of multipliers α and β for the two constraints:

$$\begin{aligned}
L = &\sum_s U_{b_s}(a_s) + \rho\overline{U}_s(z_s) - \frac{\kappa_s}{2}[(a_s - d_s)^2 + (z_s - e_s)^2] \\
&+ \sum_l \alpha_l \left(c_l - \sum_s R_{lr(s)}(a_s + x_s^0) \right) \\
&+ \sum_l \beta_l \left(c_l - \sum_s R_{lr(s)}[p_s(a_s + x_s^0) + z_s] \right) \\
= &\sum_s [U_s(a_s) - (q_{r(s)} + p_s v_{r(s)})a_s - \frac{\kappa_s}{2}(a_s - d_s)^2] \\
&+ \sum_s [\rho\overline{U}_s(z_s) - v_{r(s)} z_s - \frac{\kappa_s}{2}(z_s - e_s)^2] \\
&+ \sum_l (\alpha_l + \beta_l)c_l - \sum_s (q_{r(s)} + p_s v_{r(s)})x_s^0.
\end{aligned}$$

Here we have defined the vectors of aggregate prices per route

$$q_r = \sum_l R_{lr}\alpha_l, \quad v_r = \sum_l R_{lr}\beta_l.$$

The corresponding algorithm is defined by repeating the following two steps:

(i) For fixed d_s, e_s, run N steps of the iteration

$$\begin{aligned}
a_s :=& \arg\max_{a_s} [U_{b_s}(a_s) - (q_{r(s)} + p_s v_{r(s)})a_s \\
&- \frac{\kappa_s}{2}(a_s - d_s)^2]; \qquad (19a)
\end{aligned}$$

$$z_s := \arg\max_{z_s} [\rho\overline{U}_s(z_s) - v_{r(s)} z_s - \frac{\kappa_s}{2}(z_s - e_s)^2]; \tag{19b}$$

$$\alpha_l := [\alpha_l + \gamma_l(y_l - c_l)]^+; \tag{19c}$$

$$\beta_l := [\beta_l + \gamma_l(\tilde{y}_l - c_l)]^+. \tag{19d}$$

Here y_l, $\tilde{y}_l$ are defined in (17b-17c).

(ii) Set $d_s := a_s$, $z_s := e_s$ and go to (i).

The above algorithm is very similar to the one in Section 3. Although there are additional price and rate variables to communicate, the complexity is fundamentally the same. Solving (19a) amounts to the equation

$$U'_{b_s}(a_s) = q_s + p_s v_s + \kappa_s(a_s - d_s), \tag{20}$$

which compares bids to a linear cost; this is similar to (15) but with an additional price term $p_s v_s$ that "internalizes" the cost of the current allocation in the following auction. (19b) involves a similar calculation with the expected bids. They both have in general non-integer solutions.

4.2. Implementation and simulations

Implementing the described allocation algorithm in a real network should be possible with variants of current network protocols. For instance, reservation and price signalling between network elements can be done with the RSVP protocol, as we now briefly describe.

First, user bids are received by the brokers, where each broker is associated with a service and a route from a network access node to a server. These bids are collected until auction time.

The auction allocation is then performed following the above decentralized algorithm running in the network elements. Specifically, for the N iterations of step (i) the rate reservation variables (a_s, z_s) are sent by brokers in RSVP Path messages; prices are accumulated along a path with RSVP Resv messages in the reverse direction. The variables (d_s, e_s) are updated less frequently, and convergence is defined when variations in these variables are below a tolerance, or alternatively after a maximum number of iterations. Finally, the a_s is rounded down to a multiple of σ_s, values that are within capacity provided convergence has been attained. These circuits are reserved through a last round of RSVP reservations.

An important implementation issue is that the mean user utility function may not be known to the broker. In that case, we use an adaptive method that estimates the function $\overline{U}$ from past bids, through an exponential smoothing of the instantaneous utility function. Namely:

$$\overline{U}^{(k+1)}(z) = (1-\delta)\overline{U}^{(k)}(z) + \delta U_{b^k}(z),$$

where $\overline{U}^{(k)}$ is the current estimate. Note that this requires updating only the values of $\overline{U}$ at multiples of the circuit rate. Furthermore, the iteration applies even if the number of received bids is randomly varying in time.

This procedure allows the allocation mechanism to become independent of the bid distribution, and also of the arrival process. For instance, if bids arrive as a stationary random process (e.g. Poisson), $\overline{U}$ is well defined, but difficult to write explicitly. However, the system can estimate it through smoothing. In Fig. 8 we show an estimation example. In this case, bids arrive as a Poisson process with intensity $\lambda = 10$ bids per auction, each bid having uniform distribution in $[0,1]$. The averaging is taken over 100 auction periods with $\alpha = 0.05$. The real $\overline{U}(z)$ was calculated numerically.

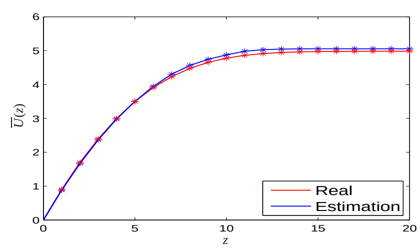

Figure 8: Estimation of the $\overline{U}(z)$ for a Poisson process

In order to evaluate the proposed algorithm, we implemented a discrete event simulator in JAVA which runs the allocation algorithm in a configurable network topology, with variable circuit demands, bid distributions and arrival processes. The simulator also implements the myopic policy and the average utility estimation presented above. We present results for three different scenarios.

4.2.1. Single link auctions.

We first compare the results of the receding horizon and myopic policies in a single link case, with 30 circuits. Auctions take place each T minutes, and bids arrive periodically with intensity λ bids/min (assumed fixed), totalling $N = \lambda T$ bids per auction.

Bids are assumed independent and uniformly distributed in $[0,1]$, and rejected bids are discarded after each auction. Accepted jobs are assumed to stay in the system an exponentially distributed time with mean 100 minutes. Hence, T is a critical system parameter: enlarging T will allow more bids to participate in a given auction and circuits to be freed in between, but a very large T will decrease the auction rate, and therefore decrease the revenue per time unit.

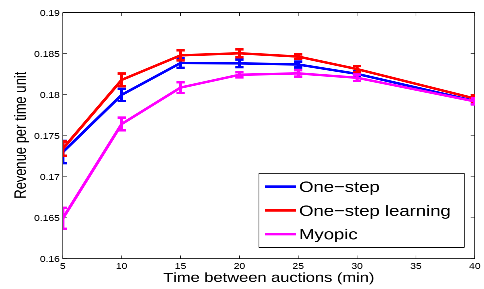

Figure 9: One link situation: 30 circuits, bid arrival rate $\lambda = 0.5$

In Fig 9 we show the results for $\lambda = 0.5$. In this figure the myopic policy is compared with the one-step ahead policy implemented with the known bids distribution and with the learning version described above. We can see that both one-step ahead policies attain more revenue per time unit than the myopic policy, as expected.

4.2.2. Linear network

We now simulate the linear network topology of Fig. 10. In this case, users in the long route 1 are expected to pay more in order to be allocated resources, since each of its circuits traverses 2 links. In order to emulate a real world situation, the bids arrive as a Poisson process of intensity λ and the learning one-step-ahead policy is used. In the first simulation, we compared the results

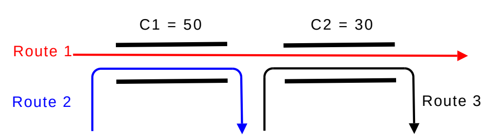

Figure 10: Linear network with varying bids.

of this policy with the myopic one by variying the bid arrival rate λ in every link and keeping the time between auctions $T = 5$ min. We fixed the mean bid of route 1 to be twice of shorter routes. Results are shown in Fig. 11, where the average income per unit time is displayed.

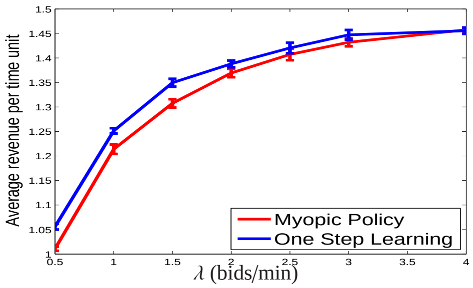

Figure 11: Linear network with varying bid arrival rate.

As we can see, also in this case the one step ahead policy attains a significative gain over the myopic policy, for a wide range of arrival rates.

Our second experiment deals with varying the mean bid over the long route. In this case, $T = 5$ min. as before and $\lambda = 1$. We assumed independent and uniform bids with mean 1 for the short routes and varying mean for the long route. Results are shown in Table 1. As

Table 1: Effect of varying the mean bid in the allocation.

$E[b_1]$	REV_1	REV_2	REV_3	a_1	a_2	a_3
0.5	0.003	0.643	0.428	0.3	47.5	28.4
1.0	0.051	0.627	0.401	2.8	45.0	25.9
1.5	0.252	0.558	0.312	9.5	38.4	19.2
2.0	0.583	0.469	0.191	17.3	30.7	11.5
2.5	0.918	0.397	0.115	22.1	25.9	6.7

· REV_s: revenue per unit time generated by service s.
· a_s: mean allocated rate in service s.

we can see, when the mean bid of broker 1 is twice as much as the others, it gets a fair share of connections. Offering more will cause most resources of link 2 to be allocated to broker 1, with broker 2 retaining its share of 10 circuits, and broker 3 will starve.

4.2.3. Overlay network.

In this final scenario, we tested the feasibility of our proposal in the more realistic situation depicted in Figure 12. In this case we have four interconnected servers and several brokers, each one attempting to secure resources of the overlay network. We have two types of demands: each connection in the short routes 1 and 3 consumes 2 circuits representing premium traffic, and the rest consume 1 circuit. The numbers over the links in Figure 12 indicate the number of available circuits. We assume that premium demand is less frequent (20%) but its mean bid is twice the bids of shorter routes.

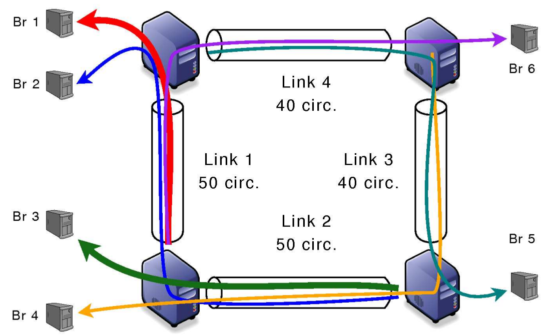

Figure 12: Overlay Network Example.

The results are shown in Table 2. We can see that the premium users who only use one link receive a substantial portion of the resources.

Table 2: Simulation results for Scenario 3

Broker	1	2	3	4	5	6
Links	1	1-2	2	2-3	3-4	4-1
REV_s	0.111	0.081	0.115	0.204	0.420	0.211
a_s	30.8	4.5	30.6	11.9	25.3	11.8

· REV_s: revenue per unit time generated by service s.
· a_s: mean allocated rate in service s.

5. Extension to multipath routing

This section considers a generalization of the previous setting, where each end-to-end service can be supported through multiple routes across the network. Therefore, for each s, instead of a single route we will allow a set of routes $\mathcal{R}(s)$. As before a broker at the edge will receive bids for each service class s, but now the allocation decision involves choosing the rates $a_{rs}, r \in \mathcal{R}(s)$ admitted in each route, for a total service rate per class

$$a_s = \sum_{r \in \mathcal{R}(s)} a_{rs}. \tag{21}$$

We now generalize the receding horizon optimization of Problem 6 to this situation. In addition to the variables a_{rs}, we define for each route the variables x^0_{rs}, z_{rs} that play the same role as before.

Problem 8. (Multipath network receding horizon allocation)

$$\max_{a_{rs},z_{rs}} \sum_s U_{b_s}\left(\sum_{r\in\mathcal{R}(s)} a_{rs}\right) + \rho\overline{U}_s\left(\sum_{r\in\mathcal{R}(s)} z_{rs}\right) \quad (22a)$$

$$\text{subject to} \quad \sum_s \sum_{r\in\mathcal{R}(s)} R_{lr}(a_{rs} + x^0_{rs}) \le c_l \quad \forall l, \quad (22b)$$

$$\sum_s \sum_{r\in\mathcal{R}(s)} R_{lr}[p_s(a_{rs} + x^0_{rs}) + z_{rs}] \le c_l \quad \forall l. \quad (22c)$$

Again we have stated this problem in its relaxed convex form, ignoring the integer constraints that would have to be imposed a posteriori by roundoff of the relaxed solution. In this regard there are two options: rounding off *each* a_{rs} to a multiple of σ_s, which means imposing each successful bidder is served through a single route, or alternatively rounding off the total a_s, assuming the network has multipath capabilities at the packet level.

Note that the objective of Problem 8 is again not strictly concave, but now in a more fundamental way, due to the sums over routes in (22a). Indeed it would be non-strict even if $U_{b_s}(\cdot)$ and $\overline{U}_s(\cdot)$ were replaced by strictly concave functions, as occurs in multipath congestion control. This was in fact what motivated the proximal optimization method in [16], and once again we will employ it here. For this purpose, introduce variables d_{rs}, e_{rs} and subtract from the objective in (22a) the quadratic term

$$\sum_s \sum_{r\in\mathcal{R}(s)} \frac{\kappa_s}{2}[(a_{rs} - d_{rs})^2 + (z_{rs} - e_{rs})^2].$$

We now describe the calculations involved in step (i) of the proximal approximation method, when one optimizes over a_{rs} and z_{rs} for fixed d_{rs}, e_{rs}. As before we will use duality, and define Lagrange multpliers α_l, β_l for the constraints at each link, updated as in (19c-19d), and the corresponding aggregate prices q_r, v_r per route. Given such prices, the update of the primal variables a_{rs}, z_{rs} is charactarized by the optimality conditions

$$U'_{b_s}\Big(\sum_{r\in\mathcal{R}(s)} a_{rs}\Big) = q_r + p_s v_r + \kappa_s(a_{rs} - d_{rs}) \quad (23)$$

$$\overline{U}'_s\Big(\sum_{r\in\mathcal{R}(s)} z_{rs}\Big) = v_r + \kappa_s(z_{rs} - e_{rs}) \quad (24)$$

Adding (23) over $r \in \mathcal{R}(s)$ and recalling (21) leads to the equation

$$U'_{b_s}(a_s) = \kappa_s a_s + \sum_{r\in\mathcal{R}(s)} [q_r + p_s v_r - \kappa_s d_{rs}],$$

from which a_s can be readily obtained. Substitution back in (23) leads to the formula

$$a_{rs} = d_{rs} + \frac{U'_{b_s}(a_s) - q_r - p_s v_r}{\kappa_s} \quad (25)$$

for the update of a_{rs}. An analogous procedure involving (24) leads to the update formula for z_{rs}.

5.1. Multipath example

In this last example we demonstrate the performance of the multipath allocation, and compare it with the single path case. Figure 13 shows both situations; as compared to the single-path topology in the top diagram, the topology in the bottom enables two paths for brokers 2 and 3.

Figure 14 shows the convergence of the proximal optimization algorithm in the multipath case for one typical auction. As we can see, the algorithm converges without oscillations.

To compare the performance, we ran 50 simulations for the single path and multipath cases. Each simulation includes 100 periodic auctions, for which we calculate the mean revenue per unit time for the four brokers. Figure 15 shows the minimum, maximum and average values over the 50 simulations for this mean revenue per unit time. As we can see, there is a clear improvement in revenue in the multipath case.

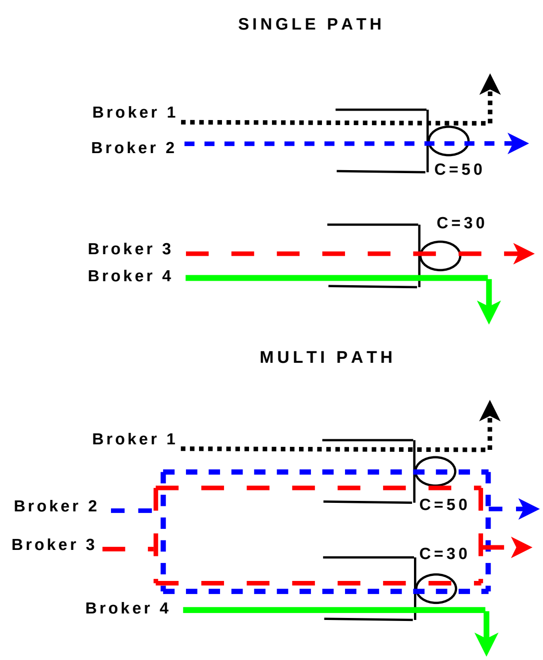

Figure 13: Single path and Multipath network topologies

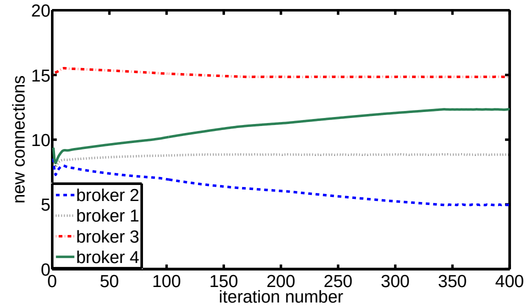

Figure 14: Multipath iteration

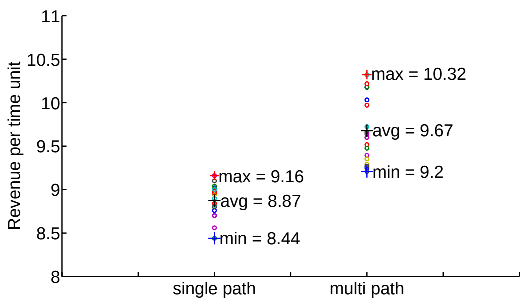

Figure 15: Revenue comparison between multipath and single path

5.2. *Considerations on the simulated algorithm and its convergence*

Simulations were performed in Matlab. There are two types of software agents, for sources and links respectively. The algorithm has an outer iteration indexed by t, and an inner one indexed by n, as follows:

(i) Fix t and the values $d_{rs}(t)$ and $e_{rs}(t)$. Initialize $q_r(t,0) := q_r(t-1,N)$ and $v_r(t,0) := v_r(t-1,N)$ (for $t = 0$, initialize at zero). For $n = 0$ to $N-1$ do

- Each source agent, independently calculates its rates $a_{rs}(t,n+1)$ and $z_{rs}(t,n+1)$ using equation (25) given $q_r(t,n)$ and $v_r(t,n)$.
- Each source s sends the rates ($a_{rs}(t,n+1)$, $z_{rs}(t,n+1)$) to the links of its routes.
- Each link accumulates the rates received and calculates using a subgradient step its new prices $\alpha_l(t,n+1)$ and $\beta_l(t,n+1)$.
- Each link agent sends the prices to the sources that use the link, and route prices are accumulated as $q_r(t,n+1)$ and $v_r(t,n+1)$.

(ii) Set $d_{rs}(t+1) = a_{rs}(t,N)$ and $e_{rs}(t+1) = z_{rs}(t,N)$.

Other than the fact that two prices and two rates must be communicated between agents, the algorithm is of the same nature as the one discussed in Section 3.2. In particular, convergence time depends on the number of steps in both the inner and outer iterations, and involves a tradeoff in the choice of κ_s in (25).

The iterations indicated in Figure 14 are the total (inner times outer) steps. We obtained convergence in around 300 iterations. Since each step is in the order of the round-trip-time (RTT), we can estimate a convergence time of around 15sec for an RTT of 50msec. This is reasonable to allocate an auction that is held periodically with a period in the order of a 5-10 minutes, which is a plausible application scenario. In more demanding scenarios, or larger RTTs, we may want to reduce the number of iterations. Beyond tuning κ_s, more significant gains would involve replacing the subgradient method of the inner iteration by something faster, e.g. Newton's method. The main challenge for such an alternative is to maintain decentralization, a feature of subgradient methods for Network Utility Maximization problems. Very recently, Wei, Ozdaglar and Jadbabaie [27] have proposed a novel distributed Newton method for NUM, that appears to make convergence 2 or 3 orders of magnitude faster than the subgradient method, while still allowing for an implementation over a network. We leave for future work the possibility that this algorithm can be adapted to our optimization situation.

6. Conclusions

In this work we proposed a mechanism for allocating network capacity through periodic auctions. We formulated the problem of maximizing operator revenue under the following constraints: the solution must be fully distributed, the network has an arbitrary topology, and the resources allocated in a given auction are reserved for the entire duration of the connection.

We formulated near-optimal policies for this problem in terms of convex optimization, through a receding-horizon version of the network utility maximization problem. Since the relevant utility is not strictly concave, we solved this problem through a proximal optimization method. Lastly, we extended this algorithm to the multipath case. All proposals lead to fully distributed solutions that can be implemented by variants of existing resource reservation protocols.

Through simulations we validated the convergence of the algorithms and the obtained performance in various network topologies. We find in particular that the receding horizon problem outperforms the myopic policy of selling all capacity in each auction, and verified the performance gains achievable through multipath. We also included practical enhancements like on-line estimation of the bid distribution. In summary, we have demonstrated a viable approach for real-time bandwidth allocation through auctions in complex distributed networks.

A. Appendix

In this appendix we obtain the expression for the threshold $\overline{w}(x)$. From (6) we have that $\overline{w}(x) = E(W(I_x)) - E(W(I_{x-1}))$ with $I_x \sim \text{Bin}(x, p)$. From the definition of W and $\overline{U}$ we can rewrite this as

$$\overline{w}(x) = E_{Ix-1}\left(\sum_{l=0}^{C-I_{x-1}} E(b^l)\right) - E_{I_x}\left(\sum_{l=0}^{C-I_x} E(b^l)\right)$$
$$= \sum_{j=0}^{x-1} A(x-1, j) \sum_{l=1}^{C-j} E(b^l) - \sum_{j=0}^{x} A(x, j) \sum_{l=1}^{C-j} E(b^l),$$

where $A(x, j) := \binom{x}{j} p^j (1-p)^{x-j}$, Operating we have

$$\overline{w}(x) = \sum_{l=C-x+1}^{C} E(b^l)(\sum_{j=0}^{C-l} A(x-1, j) - \sum_{j=0}^{C-l} A(x, j))$$
$$= \sum_{l=0}^{x-1} E(b^{C-l})(\sum_{j=0}^{l} A(x-1, j) - \sum_{j=0}^{l} A(x, j)). \quad (26)$$

Now, it can be esablished by induction that

$$\sum_{j=0}^{l} A(x-1, j) - \sum_{j=0}^{l} A(x, j) = pA(x-1, l).. \quad (27)$$

The value of the threshold in (8), follows then from (26) and (27).

Acknowledgment.

This work was partially supported by ANII-Uruguay, project PR-FCE-2009-1-2158.

References

[1] E. Altman, "Applications of Markov Decision Processes in Communication Networks: a Survey", in *Markov Decision Processes, Models, Methods, Directions, and Open Problems*, E. Feinberg and A. Shwartz (Editors) Kluwer, pp. 488-536, 2001.

[2] P. Belzarena, A. Ferragut, F. Paganini, "Auctions for resource allocation in overlay networks", *Proc. Net-Coop*, Paris, 2008.

[3] P. Belzarena, A. Ferragut, F. Paganini, "Network bandwidth allocation via distributed auctions with time reservations", IEEE Infocom 2009, Rio de Janeiro.

[4] D. Bertsekas, *Dynamic Programming : Deterministic and Stochastic Models*, Prentice-Hall, 1987.

[5] D. Bertsekas, J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, 1989.

[6] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[7] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," in *Proceedings of the IEEE*, vol. 95, no. 1, Jan 2007, pp. 255–312.

[8] C. Courcoubetis, R. Weber, *Pricing Communication Networks Economics, Technology and Modelling*, Wiley 2003.

[9] C. Courcoubetis, M. Dramitinos and G. Stamoulis, "An Auction Mechanism for allocating the bandwidth of networks to their users", *Computer Networks*, September 2007.

[10] I. Chads, M-J.Cros, F. Garcia, R. Sabbadin "Markov Decision Process Toolbox for MATLAB", `http://www.inria.fr/internet/Departements/MIA/T//MDPtoolbox/`

[11] M. Dramitinos, G. Stamoulis and C. Courcoubetis, "Auction-based resource reservation in 2.5/3G Networks" *Mobile Networks and Apps.*, 9, pp. 557-566, 2004.

[12] Z. Duan, Z-L.Zhang and Y. T. Hou, Service Overlay Networks: "SLAs, QoS and Bandwidth Provisioning", *IEEE/ACM Trans. on Networking*, pp 870 - 883, 2003.

[13] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *J. of the Operational Research Society*, 39:237–252, 1998.

[14] P. Klemperer, Auctions: Theory and Practice, The Toulouse Lectures in Economics,Princeton University Press, 2004.

[15] A. Lazar, N. Semret: Design and Analysis of the Progressive Second Price auction for Network Bandwidth Sharing. Telecommunication Systems - s. i. on Network Economics, 2000.

[16] X. Lin, N. Shroff: Utility Maximization for Communication Networks With Multipath Routing, *IEEE Trans. on Automatic Control*, Vol. 51, No. 5, pp. 766-781, May 2006.

[17] S. H. Low and D. E. Lapsley, "Optimization flow control, I: basic algorithm and convergence", *IEEE/ACM Transactions on Networking*, vol.7, no.6,pp. 861-874, December 1999.

[18] P. Maillé, B. Tuffin: Pricing the internet with multibid auctions. *IEEE/ACM Trans. on Networking.* 14(5): 992-1004 (2006)

[19] P. Maillé, B. Tuffin, "Why VCG auctions can hardly be applied to the pricing of interdomain and ad-hoc networks", Proc. NGI2007, Trondheim, Norway.

[20] A. Ozdaglar and D. Bertsekas, "Routing and Wavelength Assignment in Optical Networks", *IEEE/ACM Transactions on Networking*, Vol 11, No. 2, April 2003.

[21] M. Puterman, *Markov Decision Processes*. Wiley, N. J. 2005.

[22] R. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming",*Mathematics of Operation Research* 1 (1976), pp. 97-116.

[23] P. Reichl , S. Wrzaczek, "Equilibrium Market Prices for Multi-Period Auctions of Internet Resources". MMB 2004: 25-34.

[24] J. Shu and P. Varaiya, "Pricing network services," in Proc. IEEE INFOCOM, 2003, vol. 2, pp. 1221-1230.

[25] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhuser, Boston, MA, 2004.

[26] W. Vickrey, "Counterspeculation, auctions and competitive sealed tenders", *Journal of Finance* 16, 8-7, 1961.

[27] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A Distributed Newton Method for Network Utility Maximization," Proc. of 49th IEEE Conference on Decision and Control (CDC), Issue Date: 15-17, Dec. 2010, pp. 1816 - 1821 , 2010.

[28] N. Yamashita and M. Fukushima, "The proximal point algorithm with genuine superlinear convergence for the monotone complementarity problem", *SIAM J. on Optimization* 11 (2001), 364-379.