

Decentralized Capacity Reallocation for a Loss Network

A. Kurve¹, G. Pang² and G. Kesidis^{1,3}
¹EE, ²IE & ³CSE Depts,
 The Pennsylvania State University
 {ack205,gup3,gik2}@psu.edu

G. de Veciana
 ECE Dept,
 University of Texas, Austin
 gustavo@ece.utexas.edu

Abstract—Stochastic loss network models have been widely used to study the characteristics of a large class of resource constrained networks in terms of their blocking probability. We consider a distributed capacity allocation problem for a loss network with directed edges. Each infrastructure node (or superpeer) can independently adjust its link-capacity allocations, subject to a constraint on the total amount for each node, so as to minimize its estimate of call blocking rates. We argue via Erlang fixed-point approximation that such decentralized local changes do work to minimize a global measure of weighted call blocking rates.

I. INTRODUCTION

Loss networks provide a probabilistic framework to study resource constrained networks and have been extensively used to study a large class of diverse networks ranging from telephone networks to inventory systems in large companies. The links of the network with their finite capacities represent the limited resources of the network and are subject to utilization by users based on random service requests and random service times. Such a network is studied as a stochastic process and its stationary behavior is used to characterize its overall performance. Specifically, the stationary probability of denial of service termed as “blocking probability” is used.

However, due to the computational complexity of finding the blocking probabilities, a technique called Erlang fixed point (EFP) approximation [4] is used instead which allows for a guaranteed fixed point value for the blocking probabilities via iterative computations. An inverse loss network model can be used for capacity planning wherein the links are assigned capacities to get the desired blocking probabilities or to optimize an arbitrary cost function that reflects an application specific utility for a given topology (a recent reference is theorem 5 of [3]).

In this work we focus on the decentralized network capacity allocation or the distributed and dynamic adjustment of link capacities to deal with time-varying demand. Our model of a loss network specifically considers links that are “owned” by nodes or superpeers *i.e.*, the nodes or superpeers have the liberty to assign capacities to the links that they own subject to the constraint on the total capacity at their disposal. As an application, consider a federation of secondary providers. These providers lease a quantity of bandwidth from primary bandwidth owners, but may have flexibility regarding how

much of their leased bandwidth they allocate on particular links of the primary. Each such secondary provider must satisfy connections initiated by their own (local) clients and those initiated by (not local) others to their own clients. As another example, superpeers can adopt the following dynamic loss network model to decide how to fairly service (peer-to-peer) queries from a broader community of peers.

This makes the problem distributed in nature and necessitates the study of local decisions that help to maximize the global utility in some sense. We define a global measure of weighted call blocking rates and approximate it to allow for local decisions via Erlang fixed point approximations. We show that such a local approximation works sufficiently well in different scenarios with very few erroneous decisions which we refer to as “inconsistent” decisions in this paper.

This paper is organized as follows. We begin with the discussion on the background of loss networks and the Erlang fixed point approximation in section I. In this section we also motivate the decentralized capacity allocation problem. In section III, we explain our network model and the local approximation of the formula for finding blocking probabilities. Section IV, defines the global measure and iterative local decisions to improve the global cost. In section V, we describe the numerical experiment to study the optimality of the iterative method and the inconsistencies with local decisions and we conclude with section VI.

II. BACKGROUND

Consider a graph $G = (\mathcal{V}, \mathcal{L})$ with node (vertex) set \mathcal{V} and link (edge) set \mathcal{L} . Each link (edge) $l \in \mathcal{L}$ has a finite capacity c_l where $c_l \in \mathbb{Z}^+$ is expressed as the number of “circuits” each circuit having a fixed capacity, e.g., a circuit could be a 64 Kbps channel (voice line) or a T1 line of 1.544 Mbps. Each route $r \in \mathcal{R}$ consists of a set of links and represents a single user (source-destination pair). Each user or a route r generates “calls” at random times modeled by a Poisson process with a mean λ_r . A call setup requires one circuit from each of the links contained in the route. A call is successful if at least one circuit is available at each link belonging to the route. A call has a random life time modeled with an exponential distribution.

Many iterative improvement approaches (including exhaustive search) have been proposed for capacity allocation/planning in practical computer networking contexts, including where fault tolerance and recovery are of concern.

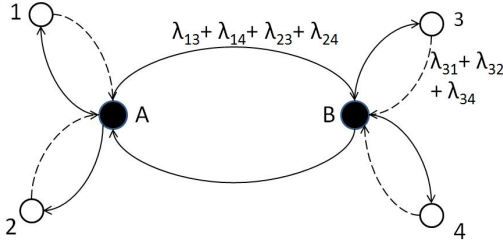


Fig. 1. Dumbbell topology example

These are typically formulated with a centralized objective function, see, *e.g.*, [13], [9] and the references therein.

In the following, we use a loss network framework. The theory of loss networks is surveyed in [5], [11] and, more recently, in [8], [14] together with a discussion of open problems. A principal idea for capacity planning in loss networks is the inverse Erlang formula wherein blocking rates are specified and suitable edge capacities can be computed to meet them for a given topology. Also, capacity planning for loss networks with time-varying workloads is considered in [1].

III. PROBLEM FORMULATION

In this paper, we formulate a decentralized problem generally similar to the mean-field framework considered in, *e.g.*, [6], [2]. Though loss networks typically work on undirected graphs, our assumption is a directed graph. The interior (infrastructure or superpeer) nodes attempt to load balance their limited resources among the peripheral nodes *requesting* paths between themselves and other peripheral nodes. Also unlike past work, we assume in the following that the total bandwidth resources allocated to each interior node, though limited, can be dynamically distributed among the physical links that connect it. That is, each interior node has a limited capacity but can allocate that capacity in different ways along its *outbound* edges.

So, the graph $G = (\mathcal{V}, \mathcal{L})$ has directed links (edges) in \mathcal{L} such that every pair of connected nodes in \mathcal{V} are connected by two links, one in each direction. Suppose each edge $l \in \mathcal{L}$ has a *single* “controlling” vertex $v \in \mathcal{V}$, denoted $l \in v$. The interior nodes are assumed to form a connected subgraph to which all other nodes are connected. Path set-up attempts follow unidirectional paths from one peripheral (non-interior) node to another.

For the dumbbell topology example of Figure 1 with six nodes, the two interior nodes are A and B and there are twelve possible paths/routes, *e.g.*, path 1, A, B, 4 from node 1 to node 4 with intensity $\lambda_{1,A,B,4} =: \lambda_{1,4}$ queries/s. Peer B controls the three (solid) edges from B to A, 3, 4. The arrival intensity on the edge from A to B is the superposition $\lambda_{1,4} + \lambda_{1,3} + \lambda_{2,4} + \lambda_{2,3}$. The solid edges have limited “circuits” $c < \infty$ and so blocking on them may occur, whereas there is no blocking on the dotted edges, *i.e.*, $c = \infty$.

Consider a loss network with a large-capacity stationary regime where the route (r) blocking probabilities B_r satisfy

$$1 - B_r \approx \prod_{l \in r} [1 - \mathcal{E}(\rho_l, c_l)] \quad (1)$$

where ρ_l is *reduced load* or the *net arrival rate* at edge l , \mathcal{E} is the Erlang blocking formula

$$\mathcal{E}(\rho, c) := \frac{\rho^c / c!}{\sum_{i=0}^c \rho^i / i!}, \quad (2)$$

$c_l \in \mathbb{Z}^+$ is the number of circuits on edge l , and we have assumed a common universal mean connection lifetime taken here to be unity without loss of generality. That is,

$$b_l := \mathcal{E}(\rho_l, c_l)$$

is the blocking probability of edge l . The net edge arrival rate can be obtained from the fixed exogenous arrival rate to each route r , λ_r , by the Erlang fixed point approximation:

$$\rho_l = \sum_{r: l \in r} \lambda_r \prod_{l' \in r, l' \neq l} (1 - \mathcal{E}(\rho_{l'}, c_{l'})) \quad (3)$$

for all $l \in \mathcal{L}$. It is known that for the above iterative computations, a unique fixed point exists. See Theorem 3.12 of [5].

In the following, we will consider dynamics in which capacities are adjusted in a decentralized and asynchronous way. We first observe from (2) that

$$\mathcal{E}(\rho, c+1) = \frac{\frac{\rho}{c+1} \mathcal{E}(\rho, c)}{1 + \frac{\rho}{c+1} \mathcal{E}(\rho, c)}, \quad (4)$$

and for $c > 1$,

$$\mathcal{E}(\rho, c-1) = \frac{\frac{c}{\rho} \mathcal{E}(\rho, c)}{1 - \mathcal{E}(\rho, c)}. \quad (5)$$

Thus, to model one such system, when c_l is increased to $c_l + 1$ for some $l \in \mathcal{L}$ and all other edge capacities do not change, the blocking probabilities \tilde{B}_r for the routes r that this link l is adjacent to become (*i.e.*, $\forall r : l \in r$):

$$\begin{aligned} 1 - \tilde{B}_r &\approx \prod_{l' \in r} (1 - \tilde{b}_{l'}) \\ &\approx (1 - \mathcal{E}(\rho_l, c_l + 1)) \prod_{l' \in r, l' \neq l} (1 - b_{l'}) \\ &= \left(1 + \frac{\rho_l}{c_l + 1} b_l\right)^{-1} \prod_{l' \in r, l' \neq l} (1 - b_{l'}). \end{aligned} \quad (6)$$

The first approximation is due to Erlang fixed point, whereas the second approximation assumes that for all edges $l' \neq l$, the change in the blocking probability $b_{l'}$ is negligible. Note that due to the change in the capacity of link l , the fixed point solution for the system is “shifted” potentially affecting the blocking probabilities of all the links and the second approximation ignores this fact. Similarly if $c_l \rightarrow c_l - 1$ for some $l \in \mathcal{L}$, then $\forall r$,

$$\begin{aligned} 1 - \tilde{B}_r &\approx \prod_{l' \in r} (1 - \tilde{b}_{l'}) \\ &\approx (1 - \mathcal{E}(\rho_l, c_l - 1)) \prod_{l' \in r, l' \neq l} (1 - b_{l'}) \\ &= \left(\frac{1 - (1 + \frac{c_l}{\rho_l}) b_l}{1 - b_l}\right) \prod_{l' \in r, l' \neq l} (1 - b_{l'}). \end{aligned} \quad (7)$$

IV. A MODEL FOR DECENTRALIZED CAPACITY CONTROL

The limited total link capacity of a interior node v is captured by

$$\sum_{l \in v} c_l = C_v, \quad (8)$$

for some fixed $C_v > 0$. Let us define the following objective function which is the total logarithmic acceptance probabilities over all routes weighed by their respective arrival rates,

$$U(\underline{c}) := \sum_r \lambda_r \log(1 - B_r), \quad (9)$$

where $\underline{c} = (c_l : l \in \mathcal{L})$, The effect of the logarithm in the expression for U is to exaggerate the smaller acceptance probabilities. Our aim is to find an optimal capacity allocation policy in order to maximize the objective function subject to the capacity constraints on all vertices (8).

Using (1), we can approximate (9) by

$$\begin{aligned} U(\underline{c}) &\approx \tilde{U}(\underline{c}) := \sum_r \lambda_r \sum_{l \in r} \log(1 - b_l) \\ &= \sum_l \log(1 - b_l) \sum_{r: l \in r} \lambda_r \\ &= \sum_l \Lambda_l \log(1 - b_l), \end{aligned} \quad (10)$$

where Λ_l is the sum of arrival rates of routes that the link l belongs to.

Suppose that there are two capacity allocation vectors \underline{c} and \tilde{c} such that there exist a single vertex v and two links $l, l' \in v$ with $c_l > 1$, $\tilde{c}_l = c_l - 1$ and $\tilde{c}_{l'} = c_{l'} + 1$, that is, one unit of capacity of vertex v is reallocated from link l to link l' . Then,

$$\begin{aligned} U(\underline{c}) - U(\tilde{c}) &\approx \tilde{U}(\underline{c}) - \tilde{U}(\tilde{c}) \\ &\approx \Lambda_l \left(\log(1 - b_l) - \log \left(\frac{1 - (1 + \frac{c_l}{\rho_l}) b_l}{1 - b_l} \right) \right) \\ &\quad + \Lambda_{l'} \left(\log(1 - b_{l'}) - \log \left(1 + \frac{\rho_{l'}}{c_{l'} + 1} b_{l'} \right)^{-1} \right) \\ &\quad \text{by (6) and (7)} \\ &= \Lambda_l \log \left((1 - b_l)^2 \left(1 - (1 + \frac{c_l}{\rho_l}) b_l \right)^{-1} \right) \\ &\quad + \Lambda_{l'} \log \left((1 - b_{l'}) \left(1 + \frac{\rho_{l'}}{c_{l'} + 1} b_{l'} \right) \right). \end{aligned} \quad (11)$$

Note that the probability (11) depends only on information that can be estimated locally at v , *i.e.*, mean link blocking rates b_l and the net arrival rate Λ_l .

We propose an approach wherein each interior node, v , selects *at random* two different edges it controls, l and l' , and decides to move one circuit from l to l' only if the quantity (11) is negative. The manner in which candidate edges to move a circuit are chosen and the fact that (11) requires only information local to v implies that this approach is decentralized. However, the tradeoff for having a decentralized scheme is the loss in accuracy due to the local approximation of the global cost function. We can easily show that $\tilde{U}(\underline{c})$ is a concave function and so there ought to be a unique maximum for the centralized constrained optimization problem. This is

TABLE I
TOTAL CAPACITY OWNED BY INTERIOR NODES

Node	Capacity	Node	Capacity
0	119	8	79
1	78	9	90
2	99	10	108
3	128	11	43
4	108	12	108
5	109	13	91
6	89	14	108
7	46		

TABLE II
ARRIVAL RATE FOR THE ROUTES

Route	1	2	3	4	5	6	7	8	9	10	11	12	13
Arrival Rate	9	11	10	8	10	10	8	10	8	9	11	11	8

due to the Erlang blocking formula $\mathcal{E}(\rho, c)$ in (2) being convex in c [12]. Thus, $\log(1 - \mathcal{E}(\rho, c))$ is concave in c . However the local approximation may not give a unique solution or a guaranteed increase in the cost for each reallocation step. In the numerical experiments below, we explore this phenomenon along with other issues such as speed of convergence and the optimality with respect to \tilde{U} of the distributed solution.

V. NUMERICAL EXPERIMENTS

We numerically studied the effect of local decisions on the global cost function equation (9) and now give the results for a typical topology we considered. Among several others, we simulated a network of 15 interior nodes using NetLogo [10] whose topology is depicted in Figure 2. Each interior node had a random number (ranging from 1 to 4) of directed links originating from itself to another randomly chosen node other than itself. We created 13 random routes each with a random arrival rate. In order that the Erlang fixed point (EFP) approximation is reasonably accurate, the initial capacity of each link was chosen so as to get the average ‘‘incident’’ load on each link, *i.e.*, the sum of the *unblocked* arrival rates of all the routes that use a link divided by its capacity, between 0.9 to 1, *i.e.*, a heavy traffic regime when EFP approximation is accurate [5] and motivating capacity reallocation. Each node owned links that originated from itself, *i.e.*, the node could arbitrate the assignment of capacity to the links that it owned subject to the sum of their capacities remaining constant. The fixed total capacities per interior node (in discrete circuit units) is given in Table I.

Following the initial setup, we calculated the net arrival rate for each link and the blocking probability by using the EFP recursion. For this initial capacity assignment, we calculated the objective function given by equation (9). We then performed iterative, decentralized, asynchronous capacity adjustment as described below: Two directed edges owned by a randomly chosen node were chosen at random, the value of $\tilde{U}(\underline{c}) - \tilde{U}(\tilde{c})$ using (11) was calculated, and a circuit was transferred between the chosen links if this value was negative. For purposes of performance evaluation, the new blocking

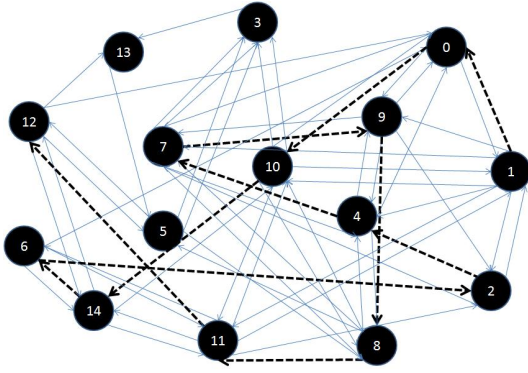


Fig. 2. Typical assessed network topology with one of its 13 routes indicated in bold.

probabilities were then calculated using the EFP recursion for the new capacity allocation vector before the next local capacity adjustment iteration was executed. The steps are summarized in Algorithm 1.

Algorithm 1 Steps for Simulation

- 1: initialize random graph
- 2: create random routes with random λs
- 3: calculate b_l and ρ_l using EFP
- 4: **repeat**
- 5: $d \leftarrow 0$
- 6: **repeat**
- 7: randomly choose a node with links > 2
- 8: randomly choose 2 links owned by the above node
- 9: Calculate $d = \tilde{U}(\underline{c}) - \tilde{U}(\underline{\tilde{c}})$
- 10: **until** $d < 0$
- 11: reallocate capacities for the above two links
- 12: calculate b_l and ρ_l using EFP
- 13: calculate \tilde{U} with this new allocation
- 14: **until** $iterations < max_iterations$
- 15: Plot \tilde{U} versus iterations

We repeated this 250 times *i.e.*, $max_iterations = 250$, each time observing the change in $\tilde{U}(\underline{c})$. This was done for 10 different initializations. We saw that the objective function predominantly increased with each local capacity reallocation, except on few occasions when there was an inconsistency in the local decision *i.e.*, for such reallocation steps the global cost function actually decreased. These inconsistent decision are due to approximation for deriving local cost given by (11). In Figure 3, a graph is shown of the objective function \tilde{U} versus the number of iterations for three different initial values of feasible capacity allocations, *i.e.*, sum of capacities of the links originating from the same node remains constant (as shown in Table I) for all the initializations. Thus, we can conclude that incremental decentralized capacity reallocation indeed improves the system performance. Moreover, we observed that for all these initial capacity allocations, the final link-capacity vectors were very close to each other, the maximum L_1 distance¹ between them being less than 10 circuits. The algorithm often did not converge, but orbited in

¹The L_1 distance between two link-capacity vectors is $|\underline{c} - \underline{c}'| = \sum_l |c_l - c'_l|$ circuits.

TABLE III
PROPORTIONAL CAPACITY ALLOCATION

Node	Link	Λ_l (circuits* per unit time)	Final capacity (circuits)
0	(0,1)	18	19
	(0,2)	9	12
	(0,5)	10	12
	(0,6)	30	31
	(0,13)	38	36
	(0,14)	8	9
10	(10,3)	19	21
	(10,4)	18	19
	(10,5)	8	9
	(10,7)	39	39
	(10,11)	19	20

* service requests

this group of final link-capacity vectors. Again, this was due to the approximation errors of (11). If we slightly changed the local computation algorithm so that we swapped a link only when $\tilde{U}(\underline{c}) - \tilde{U}(\underline{\tilde{c}}) < -\delta$, where δ is a reasonably small positive value such as 0.01, then the algorithm converged to one a “nearly optimal” link-capacity vector. However, due to the discrete, incremental and probabilistic nature of the improvement steps, the algorithm did not necessarily converge to the same final values for the different initializations. We repeated the experiment for 10 different initial feasible capacity vectors. The average number of iterations with occurrence of inconsistent decision (again, where the change in $\tilde{U}(\underline{c})$ predicted by our approximation as positive is actually negative according to the EFP approximation) was about 12 out of 250 iterations for each initialization of capacity vector. The maximum observed value of $\tilde{U}(\underline{c})$ was -38.24 . The minimum L_1 distance between any pair of initial capacity allocation vector was 162 circuits. The minimum L_1 distance between the initial and final capacity vectors of a single trial was 126 circuits.

It is observed that for the optimal capacity allocation, the capacity assigned to a link l is approximately proportional to Λ_l *i.e.*, the sum of arrival rates of routes that use the link l . As an example, Table III shows the capacity and the total arrival rate for the links for two nodes 0 and 10 for the same realization of randomly generated network as above.

A. Closeness to Optimality

We know that finding the solution to the original global cost given by (9) is known to be $\sharp P$ -complete problem [7]. So instead we compare the solution of our method with the optimal solution for the global cost (10) which can be calculated via centralized iterative improvement. Again, due to the concavity of (10), we are guaranteed to find the global optimum. To do so, we need to slightly modify Algorithm 1 and replace the calculation of $\tilde{U}(\underline{c}) - \tilde{U}(\underline{\tilde{c}})$ with EFP computation to decide whether to transfer the circuit. With this procedure, the value of the cost function (10) was found to be -38.054 for the experiment described above which is very close to the values found by our decentralized scheme. The L_1 distance between the globally optimal solution and

TABLE IV
INCONSISTENT DECISIONS FOR ASYNCHRONOUS REALLOCATION

No. of nodes simultaneously deciding	1	2	3	4	5
No. of inconsistent decisions	15	23	42	59	67
Iterations needed	267	245	211	183	162

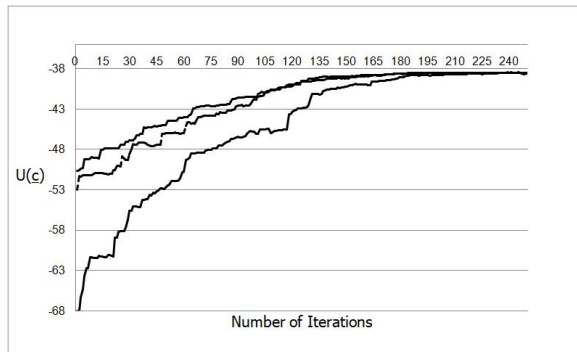


Fig. 3. Global objective with iterative capacity adjustments

the best solution given by our method, out of the 10 different initializations, was just 5 circuits.

B. Parallel local decisions

The decentralized scheme we have discussed above allows for only a single node to make decisions at a given iteration. In our next experiments, we allowed for more than one node to make a decision at a given iteration. As shown in Table IV, we observed that the average number of inconsistent decisions increase with the number of nodes that simultaneously decided to reallocate circuits. Also, the average number of iterations to get to 0.1% of the optimal solution decreased with increased parallelism. Again, in this setup the nodes were chosen randomly for parallel reallocation.

For our next experiment, we modeled a graph that had two components which were densely connected within themselves and connected sparsely to each other by one link in each direction. The routes were randomly generated by a method that resembled random walk along the graph. The sparse connectivity implied that the probability that any two links in the different components share a route is small. We allowed for parallel moves by nodes, one in each component, and observed the number of inconsistent decisions which increased only slightly from 27 when one node made a move per iteration to 32 for the parallel moves as described above. Thus, one might choose to select nodes which are geodesically far from each other for parallel moves to alleviate the penalty in terms of inconsistent decisions.

VI. CONCLUSION

We proposed a decentralized capacity reallocation scheme for a loss network model where nodes arbitrate capacity over the links they “own” in order to optimize a global measure of system performance. We studied the inconsistent decisions in the local decisions with respect to the improvement of the global measure U for different scenarios. It was observed that such a distributed scheme gave a nearly optimal solution with reasonably small number of inconsistent decisions (where U

did not increase). We can infer from our experiments that one can allow for parallel moves only by “distant” nodes in order to minimize the number of inconsistent decisions.

REFERENCES

- [1] S. Bhadra, Y. Lu, and M.S. Squillante. Optimal capacity planning in stochastic loss networks with time-varying workloads. In *Proc. ACM SIGMETRICS*, San Diego, 2007.
- [2] M. Huang, P.E. Caines and R.P. Malhame. Social certainty equivalence in mean field LQG control: social, Nash and centralized strategies. In *Proc. 19th Mathematical Theory of Networks and Systems (MTNS)*, Budapest, July 2010.
- [3] K. Jung, Y. Lu, D. Shah, M. Sharma, and M.S. Squillante. Revisiting stochastic loss networks: structures and algorithms. In *Proc. ACM SIGMETRICS*, Annapolis, MD, 2008.
- [4] F.P. Kelly. Blocking probabilities in large circuit-switched networks. *Advances in Applied Probability*. **1**(3): 473–505, 1986.
- [5] F.P. Kelly. Loss networks. *Annals of Applied Probability*. **1**(3): 319–378, 1991.
- [6] F.P. Kelly, A.K. Maulloo and D.K.H. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society* **49**: 237–252, 1998.
- [7] G. Louth, M. Mitzenmacher and F. Kelly. Computational complexity of loss networks. *Theoretical Computer Science* vol. 125, pages 45–59, 1994.
- [8] R.R. Mazumdar. *Performance Modeling, Loss Networks, and Statistical Multiplexing*. Morgan & Claypool Publ., San Francisco, 2009.
- [9] D. Medhi and D. Tipper. Some Approaches to Solving a Multihour Broadband Network Capacity Design Problem with Single-path Routing. *Telecommunication Systems* **13**(2–4), pp. 269–291, 2000.
- [10] NetLogo itself: Wilensky, U. 1999. NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.
- [11] K.W. Ross. *Multiservice Loss Models for Broadband Telecommunication Networks*. Springer-Verlag, London, 1995.
- [12] W. Whitt. The Erlang B and C formulas: problems and solutions. *Unpublished manuscript*. <http://www.columbia.edu/~ww2040/ErlangBandCFormulas.pdf>
- [13] K. Wu and D. S. Reeves. Capacity planning of DiffServ networks with best-effort and expedited forwarding traffic. In *Proc. IEEE ICC*, Anchorage, May 2003.
- [14] S. Zachary and I. Ziedins. Loss networks. Technical Report arXiv:0903.0640v1, Mar. 3, 2009. Available at <http://www.arxiv.org>