1

# On-line Network Routing – A survey

Amos Fiat [†] Stefano Leonardi [‡§]

## 1 Introduction

The problems associated with routing communications on a network come in a large variety of flavors. Many of these problems would be uninteresting if link bandwidth was unlimited and switching speeds were imperceivable.

One often distinguishes between *circuit routing* and *packet routing*. The main difference between the two is that packet routing allows one to store transmissions (packets) in transit and forward them later whereas circuit routing as it is conceived does not. In fact, one can today perform streaming video (a virtual circuit) on IP (a packet routing protocol) using ATM virtual circuits that are themselves transmitted as packets, so confusion is inherently unavoidable. To simplify our discussion, in this survey we deal with idealized models for circuit routing and packet routing.

A technological advance with significant consequences regarding routing is the use of optical fiber and optical switching elements. We will distinguish between electrical routing (the "standard model") and optical routing, the two models are quite different. Optical routing itself has several variant models, depending on the technological assumptions used. These differing assumptions range from available hardware to futuristic assumptions about what could possibly be done, all these assumptions are motivated by wavelength division multiplexing [Gre92].

One obviously has to define the goal of a routing algorithm, *i.e.*, the objective function by which its performance will be measured. One possible objective is to reduce the *load* in the network, e.g., route calls so that the maximal load on any link is minimized. Load can be taken to mean different things in different contexts. In electrical routing networks, load could be taken to mean the utilized bandwidth on a link compared with the maximal link capacity, *i.e.*, the percent utilized. Reducing the load in this context means reducing the highest percent utilized, over all links. In the context of optical routing, load could be taken to mean the number of different wavelengths in use in the network. Reducing the load means routing the same calls while using less wavelengths.

We will present competitive online algorithms whose goal is to reduce the load defined as percent utilization of a link, or number of wavelengths required in an optical network. *I.e.*, algorithms whose load requirements will never be more than $c$ times the load requirements of an optimal solution, where $c$ is the appropriate competitive ratio.

A potential problem with the load measure is that nothing prevents the load requirements to exceed the (real) available resources. As long as no load requirement every exceeds 100% the available resources, we're fine. Once the load requirement exceeds 100% of the available resources (link bandwidth, or wavelength

[*]This survey will appear as a chapter of the book "On-line Algorithms", A. Fiat and G. Woeginger eds, Springer-Verlag, 1998.

[†]Department of Computer Science, Tel Aviv University, Tel Aviv. Research supported in part by two grants from the Israel Academy of Sciences. e-mail: fiat@math.tau.ac.il

[‡]This work was done while the author was visiting the Max-Planck Institute für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany.

[§]Dipartimento di Informatica Sistemistica, Università di Roma "La Sapienza", via Salaria 113, 00198-Roma, Italia. This work was partly supported by EU ESPRIT Long term Research Project ALCOM-IT under contract n. 20244, and by Italian Ministry of Scientific Research Project 40% "Algoritmi, Modelli di Calcolo e Strutture Informative". email: leon@dis.uniroma1.it

capacity of optical fiber) this measure becomes problematic.

Obviously, whenever the resources required to perform some task exceed the available resources, something has to give. One possible interpretation of a routing algorithm that uses load greater than actually available is that the bandwidth allocated to every relevant call goes down, a slowdown is introduced and rather than give a call the bandwidth requested, the quality of service goes down.

In some cases this might make sense, but certainly not when performance guarantees are required (*e.g.*, streaming video). An alternative to reducing the quality of service is to reject (some) communication requests yet give others (those accepted) the quality of service (*e.g.*, bandwidth) they require.

The *throughput* objective function seeks to maximize the benefit derived from accepting communication requests, while remaining within the constraints of the available resources. Benefit can be various functions of the communication requests, such as a dollar value attached to every such request, a function of the geographical distance, etc. The primary benefit function considered in the algorithms described herein is the total number of communication requests serviced. *I.e.*, a benefit of one for every call accepted.

Generally, one can distinguish between *cost problems* where the goal of the algorithm is to reduce the cost of dealing with some sequence of events $\sigma$, and *benefit problems* where the goal of the algorithm is to increase the benefit associated with dealing with some sequence of events $\sigma$. The competitive ratio of an online algorithm $ON$ for a cost problem is defined to be

$$\sup_{\sigma} \frac{\mathrm{Cost}_{\mathrm{ON}}(\sigma)}{\mathrm{Cost}_{\mathrm{OPT}}(\sigma)},$$

while for a benefit problem is defined to be

$$\sup_{\sigma} \frac{\mathrm{Benefit}_{\mathrm{OPT}}(\sigma)}{\mathrm{Benefit}_{\mathrm{ON}}(\sigma)},$$

where $\mathrm{ON}(\sigma)$ and $\mathrm{OPT}(\sigma)$ denote the on-line and the optimal solution for a sequence $\sigma$. When considering randomized algorithms, the definition of competitive ratio is in terms of the expected cost or benefit of the algorithm.

The paging problem is a cost problem, as are the load variants of the various routing problems. The throughput variants are benefit problems. Note that we adapt the convention that the competitive ratio is always greater or equal to one.

A typical throughput problem we would like to deal with is as follows:

Given a network, $G = (V, E)$, with link capacities $u : E \mapsto R$, the routing algorithm receives requests of the form: create a virtual circuit from $v \in V$ to $w \in V$ for a duration of $d$ time units and of bandwidth $b$. The algorithm can do one of the following:

1. Assign the virtual circuit request (call) a route in the network $v, x_1, x_2, \ldots, x_k, w$ such that $(v, x_1) \in E$, $(x_i, x_{i+1}) \in E$, $(x_k, w) \in E$, and the utilized capacity of every one of these edges, including the new call, does not exceed the edge capacity.

2. Reject the call.

The algorithm that deals with the problem above should try to ensure that accepting the call and routing it via the route chosen will not create serious problems in the future in that it will make many other calls impossible to route. Because of the nature of the competitive ratio, what we really care about is that the set of calls to be accepted should be close (in benefit) to the set of calls accepted by the all powerful adversary.

Accepting and rejecting calls, if allowed in the model, is also known as *Call control* or *Admission control*. In some special networks *e.g.*, a single link from $A$ to $B$, there are no real "routing" decisions to be made. All calls between $A$ and $B$ make use of the (single) link between them. Here, call control is the only consideration. However, we will call all these problems jointly routing problems, whether route selection is an issue or not, and whether call control is allowed or not.

Throughput competitive routing algorithms on electrical networks can be considered as belonging to one of two categories:

1. Low bandwidth requirement: all calls require no more than a small fraction of the minimal link capacity. Assuming the low bandwidth requirement allows us to present a competitive algorithm for any network topology. This algorithm is deterministic.
2. High bandwidth requirement: otherwise. Deterministic algorithms cannot work well on high bandwidth calls in any network topology. Randomized algorithms give good competitive ratios for some topologies, but not all.

There are a great many other issues and models that have been studied in the context of competitive routing. We will at least try to mention such work.

Despite the vast number of models and results, we can isolate certain combinatorial problems that underly many of the problems. The load problems on electrical networks and throughput competitive algorithms for low bandwidth calls are intimately related to the problem of multicommodity flow [LMP+91]. The high bandwidth throughput competitive algorithms, the load problems on optical networks, and the throughput problems on optical networks are all strongly connected to the problem of finding edge disjoint paths in a graph [Fra]. Optical routing, almost by definition, is related to the problem of coloring paths in graphs.

### 1.1 Structure of this paper

Algorithms for electrical networks, and in particular for the virtual circuit routing problem, are covered in Section 2. In this section we will present competitive algorithms for the load balancing version and for the throughput version with small bandwidth requirement, and for the throughput version with high bandwidth requirement on specific topologies. Optical routing is studied in Section 3.

## 2 On-line virtual circuit routing

Consider a sequence $\sigma = \{(s_j, t_j)\}$ of pairs of nodes in a network $G = (V, E)$ of $|V| = n$ vertices and $|E| = m$ edges. Each pair $(s_j, t_j)$, also said call in the following, asks for the establishment of a virtual circuit on a path connecting $s_j$ to $t_j$ with transmission rate $r_j$ for a duration $d_j$. Each link $u \in E$ of the network has a maximum available bandwidth $u(e)$. Each call is processed before any future call is known.

This problem, known as the *call control* problem, was first studied from the point of view of competitive analysis by Garay and Gopal [GG92] and by Garay, Gopal, Kutten, Mansour and Yung [GGK+93]. The attention for this problem was motivated by the emerging of the ATM protocol, where a guaranteed quality of service is enforced by reserving the required bandwidth on a virtual circuit connecting the communicating parties on which the whole information stream is sent.

A throughput version and a load balancing version of the problem have been studied.

In the *throughput* version each pair of vertices $(s_j, t_j)$ is associated with a benefit $b_j$. For each call two kinds of decisions are involved. *Admission control*, i.e., the choice of whether to accept the call or not, and *route selection*, i.e., the decision on the route on which to establish the virtual circuit of an accepted call. These decisions must obey the constraint that the sum of the transmission rates of the virtual circuits that cross a given edge do not exceed the capacity of that edge. The goal is to maximize the overall benefit obtained from the sequence of calls.

In the *load balancing* version all the pairs of the sequence must be connected with a virtual circuit. The load of an edge is defined as the ratio between the sum of the transmission rates of the virtual circuits that include that edge and the capacity of the edge. The goal is to minimize the maximum load of an edge.

The throughput version of the problem turns out to be considerably easier if the transmission rate of any call is always small when compared with the minimum capacity of an edge. For this case deterministic on-line algorithms that work for any network topology have been designed and proved to achieve a good competitive ratio [AAP93].

The algorithm for the load balancing version and for the throughput version with small bandwidth requirement are both applications of the idea of associating with edges a cost that is exponential in the fraction of the capacity of that edge already assigned [AAF$^+$93]. For this reason we will present these two algorithms and other direct applications of the exponential function altogether in Section 2.1. (The exponential function also finds applications to on-line load balancing on parallel machines. See Chapter **??**).

As mentioned before, the nature of the problem changes completely if the transmission rate requested from calls is not a small part of the minimum capacity of an edge. An appealing aspect of this case is that in its basic form (when link capacities are 1, call requests bandwidth 1, and duration are infinite), it is an on-line version of the extensively studied combinatorial problem of the maximization of the number of edge-disjoint paths. In this case deterministic algorithms can achieve only a very poor competitive ratio even for simple network topologies like lines or trees. In Section 2.2 we will describe the results obtained by randomized algorithms for several network topologies. The idea of using an exponential function actually plays a central role also in this context, as a matter of facts that some of these algorithms use variations of the algorithm for the throughput version with small bandwidth requirement as a subroutine. However, there is no randomized on-line algorithm that is efficient for all the network topologies, since there are specific topologies where also randomized on-line algorithms achieve a very poor competitive ratio [BFL96].

Many variations of the basic virtual circuit routing problem have been studied. Other network problems have also been studied using a similar approach. We will try to take into account of this work in Section 2.3 and in the section where we describe the use of the exponential function.

## 2.1 The applications of the exponential function

An on-line algorithm based on the use of an exponential cost function has been first presented by Aspnes, Azar, Fiat, Plotkin and Waarts [AAF$^+$93] for the load balancing version and by Awerbuch, Azar and Plotkin [AAP93] for the throughput version with small bandwidth requirement of virtual circuit routing. We denote in the following this last algorithm by $AAP$. The idea is that of associating with each edge of the network a cost that is exponential in the fraction of the capacity of that edge assigned to on-going circuits. We describe an algorithm based on this idea and show how it is applied to both load balancing and throughput.

For sake of simplicity, we restrict the description to the case where calls are permanent, or equivalently have infinite duration. Moreover, the benefit $b_j$ of call $j$ is assumed without loss of generality in the range $[1, B]$. We indicate with $\mathcal{A}$ the set of calls accepted by the on-line algorithm and with $\mathcal{A}^*$ the set of call accepted by the optimal off-line algorithm but rejected by the on-line algorithm. (Observe that for the load balancing version no call is rejected.) If the $j$th call is accepted, let $P(j)$ and $P^*(j)$ be the path on which the virtual circuit of call $j$ is established by the on-line and by the optimal algorithm.

We denote with $l_e(j) = \frac{\sum_{j\in\mathcal{A}, i<j: e\in P(j)} r_j}{u(e)}$ and with $l^*_e(j) = \frac{\sum_{j\in\mathcal{A}^*, i<j: e\in P^*(j)} r_j}{u(e)}$ the on-line and the optimal load on edge $e$ when the $j$-th call of the sequence is presented. Let $\Lambda = \max_{e\in E} l_e(f)$ and $\Lambda^* = \max_{e\in E} l^*_e(f)$ the on-line and the optimal maximum load on a link of the network at the end of the sequence. The maximum load is 1 in the throughput version since the capacity of edges cannot be exceeded, while it is the objective function to minimize in the load balancing version.

Let $\mu$ be a parameter to be specified later. Define by $c_e(j) = \mu^{\frac{l_e(j)}{\Lambda}}$ the exponential cost of edge $e$ when call $j$ is presented. The cost of a path $P(j)$ that connects $s_j$ to $t_j$ is defined as

$$C(j) = \sum_{e\in P(j)} \frac{r(j)}{u(e)} c_e(j).$$

The algorithm for the $j$th call of the sequence is the following:

Let $P^{ON}(j)$ be the path with cost $C^{ON}(j) = \min_{P(j)} C(j)$.

If $C^{ON}(j) \leq 2mb_j$ then accept call $j$ on path $P^{ON}(j)$, else reject call $j$.

**Load balancing.** We describe how the algorithm achieves for load balancing on any network topology a maximum load that is away from the optimum maximum load for at most an $O(\log n)$ factor. Assume for the moment that the algorithm knows the optimal load $\Lambda^*$. We show that if we assign $\mu = 2m$, $\Lambda = 2\Lambda^* \log \mu$, then the algorithm does not reject any call and the load on each edge is at most $\Lambda$.

Consider the time at which call $(s_j, t_j)$ is presented. The cost of any path $P(j)$ connecting $s_j$ to $t_j$ is at most $Z(j) = \sum_{e \in E} c_e(j)$, the sum of the exponential costs of all the edges of the network when call $j$ is presented. We prove in the following that $Z(j) \leq 2m$, subject to accept each call $i < j$ on a minimum cost path $P^{ON}(i)$. This immediately implies:

$$\mu^{\frac{l_e(j)}{\Lambda}} \leq Z(j) \leq 2m,$$

and then $l_e(j) \leq \Lambda$.

We use the following potential function to prove the claim :

$$\Phi(j) \quad = \quad \sum_{e \in E} c_e(j)(1 - \frac{l_e^*(j)}{2\Lambda^*}).$$

We have $Z(j) \leq 2\Phi(j) \leq 2(\Phi(j) - \Phi(0)) + 2m$. $Z(j)$ is then bounded by $2m$ if $\Phi$ does not increase when a call $i < j$ is accepted. This is proved as follows:

$$\begin{aligned}
\Phi(i+1) - \Phi(i) \quad &\leq \quad \sum_{e \in P^{ON}(i)} (\mu^{\frac{l_e(i+1)}{\Lambda}} - \mu^{\frac{l_e(i)}{\Lambda}}) \\
&- \quad \frac{1}{2\Lambda^*} \sum_{e \in P^*(i)} (\mu^{\frac{l_e(i+1)}{\Lambda}} l_e^*(i+1) - \mu^{\frac{l_e(i)}{\Lambda}} l_e^*(i)) \\
&= \quad \sum_{e \in P^{ON}(i)} (\mu^{\frac{l_e(i) + \frac{r_i}{u(e)}}{\Lambda}} - \mu^{\frac{l_e(i)}{\Lambda}}) \\
&- \quad \frac{1}{2\Lambda^*} \sum_{e \in P^*(i)} (\mu^{\frac{l_e(i) + \frac{r_i}{u(e)}}{\Lambda}} (l_e^*(i) + \frac{r_i}{u(e)}) - \mu^{\frac{l_e(i)}{\Lambda}} l_e^*(i)) \\
&\leq \quad \sum_{e \in P(i)} (\mu^{\frac{r_i}{u(e)\Lambda}} - 1)\mu^{\frac{l_e(i)}{\Lambda}} - \frac{1}{2\Lambda^*} \sum_{e \in P^*(i)} \frac{r_i}{u(e)} \mu^{\frac{l_e(i)}{\Lambda}} \\
&\leq \quad \sum_{e \in P(i)} (2^{\frac{r_i \log \mu}{2u(e)\Lambda^* \log \mu}} - 1)\mu^{\frac{l_e(i)}{\Lambda}} - \frac{1}{2\Lambda^*} \sum_{e \in P^*(i)} \frac{r_i}{u(e)} \mu^{\frac{l_e(i)}{\Lambda}} \\
&\leq \quad \frac{1}{2\Lambda^*} \sum_{e \in P(i)} \frac{r_i}{u(e)} \mu^{\frac{l_e(i)}{\Lambda}} - \frac{1}{2\Lambda^*} \sum_{e \in P^*(i)} \frac{r_i}{u(e)} \mu^{\frac{l_e(i)}{\Lambda}} \\
&\leq \quad 0.
\end{aligned}$$

The last two equations follow since $r_i \leq u(e)$, $2^x - 1 \leq x$ if $0 \leq x \leq 1$, and $P^{ON}(i)$ is the minimum cost path.

To complete the description of the result we are only left to remove the assumption that the algorithm knows the optimal load $\Lambda^*$. This is done [AAF+93] by using a doubling technique based on the observation that if the algorithm exceeds load $\Lambda$ then we do not have a correct estimation of $\Lambda^*$. We initially run a copy of the algorithm with $\Lambda^*$ equal to the ratio between the transmission rate of the first call and the maximum capacity of an edge. Each time the algorithm exceeds the maximum load, we double the estimation of $\Lambda^*$ and start a new copy of the algorithm. This results in a multiplicative factor of 4 in the competitive ratio of the algorithm.

The $O(\log n)$ competitive ratio is the best achievable up to a constant factor. This lower bound has been shown in [AAF+93] for a directed network. Bartal and Leonardi [BL97] shows a randomized lower bound for an undirected mesh.

The algorithm can be extended to calls with limited duration. If $D$ is the ratio between the maximum and the minimum duration, an $O(\log nD)$ competitive deterministic algorithm is presented in [AKP+93]. In this algorithm a distinct cost is associated with a link for each unit of time, to model the assignment of a link to different circuits for different units of time. A matching $\Omega(\log nD)$ lower bound for this algorithm is not known.

The problem changes dramatically if the duration of the call is not known in advance. In this case an $\Omega(\sqrt{n})$ randomized lower bound is proved by Azar, Broder and Karlin [ABK92] for load balancing on parallel machines and extends to virtual circuit routing. This lower bound is proved on a sequence with exponentially growing holding times and then it does not exclude the existence of an $O(\log D)$ competitive algorithm. An $\Omega(\min(n^{1/4}, D^{1/3}))$ randomized lower bound has been later presented by Ma and Plotkin [MP97]. A matching $O(\sqrt{n})$ upper bound is proved by [AKP+93].

**Throughput.** We prove in the following the competitive ratio of the algorithm for the throughput version. We seek for maximizing the benefit of the algorithm assuming the following restriction over the ratio between the transmission rate of a call and the bandwidth of a link of the network:

$$\frac{1}{P} \leq \frac{r_j}{u(e)} \leq \frac{1}{\log \mu}. \tag{1}$$

We assume in the algorithm $\Lambda = 1$ and $\mu = 4mPB$.

We first show that when a new call $j \in \mathcal{A}$ is accepted by the algorithm, the capacity of the edges is not exceeded.

**Lemma 1** *The solution given by the algorithm is feasible.*

**Proof.** Assume by contradiction that the capacity of edge $e \in P^{ON}(j)$ is violated for the first time when call $j \in \mathcal{A}$ is accepted. Since the solution was feasible when call $j$ was presented, and $\frac{r_j}{u(e)} \leq \frac{1}{\log \mu}$, we have that $l_e(j) > 1 - \frac{1}{\log \mu}$. Then, for $C^{ON}(j)$ it holds

$$C^{ON}(j) \quad > \quad \frac{r(j)}{u(e)} \mu^{1-\frac{1}{\log \mu}} \geq \frac{1}{P} \frac{\mu}{2} = 2mB,$$

a contradiction, since call $j$ has been accepted. ■

The next lemma gives an upper bound over the benefit obtained by the optimal algorithm on calls of $\mathcal{A}^*$, as function of the sum of the exponential costs of the edges in the network at the end of the sequence. We indicate with $c_e(f)$ the cost of edge $e$ at the end of the sequence.

**Lemma 2** $\sum_{j \in \mathcal{A}^*} b_j \leq \frac{1}{2m} \sum_{e \in E} c_e(f)$.

**Proof.** Any call $j \in \mathcal{A}^*$ has been rejected by the on-line algorithm. We can then write:

$$\sum_{j \in \mathcal{A}^*} b_j \quad < \quad \frac{1}{2m} \sum_{j \in \mathcal{A}^*} \sum_{e \in P^*(j)} \frac{r_j}{u(e)} c_e(j)$$

$$\begin{aligned}&\leq \frac{1}{2m}\sum_{e\in E} c_e(f) \sum_{j\in \mathcal{A}^*: e\in P^*(j)} \frac{r_j}{u(e)}\\&\leq \frac{1}{2m}\sum_{e\in E} c_e(f),\end{aligned}$$

where the last equation follows since the capacity of edges cannot be exceeded.

■

The next lemma gives a lower bound on the benefit obtained by the on-line algorithm.

**Lemma 3** $\frac{1}{2m}\sum_{e\in E} c_e(f) \leq (1+\log\mu)\sum_{j\in\mathcal{A}} b_j$.

**Proof.** It is easy to verify that it is sufficient to prove for any call $j \in \mathcal{A}$, $\sum_{e\in P^{ON}(j)}(c_e(j+1) - c_e(j)) \leq 2mb_j \log\mu$. This follows from:

$$\begin{aligned}\sum_{e\in P^{ON}(j)} (c_e(j+1) - c_e(j)) &= \sum_{e\in P^{ON}(j)} (\mu^{l_e(j)+\frac{r_j}{u(e)}} - \mu^{l_e(j)})\\&= \sum_{e\in P^{ON}(j)} \mu^{l_e(j)}(2^{\log\mu\frac{r_j}{u(e)}} - 1)\\&\leq \log\mu \sum_{e\in P^{ON}(j)} \frac{r_j}{u(e)}\mu^{l_e(j)}\\&\leq 2mb_j\log\mu,\end{aligned}$$

where the third inequality follows since $2^x - 1 \leq x$ if $0 \leq x \leq 1$, and the fourth inequality stems from the fact that call $j$ has been accepted by the on-line algorithm.

■

We then conclude:

**Theorem 4** *The algorithm for the throughput version of on-line virtual circuit routing is $O(\log\mu)$ competitive.*

**Proof.** Denote with $OPT$ and with $ON$ the optimal and the on-line benefit over the sequence. The benefit of the optimal algorithm $OPT$ is upper bounded by $\sum_{j\in\mathcal{A}\cup\mathcal{A}^*} b_j$, while the on-line benefit is $ON = \sum_{j\in\mathcal{A}} b_j$. Combining Lemmas 2 and 3 we derive $OPT \leq (2+\log\mu)\sum_{j\in\mathcal{A}} b_j \leq (2+\log\mu)ON$, then proving the theorem. ■

The competitive ratio of an algorithm for this problem can actually be improved to $O(\log L)$, where $L$ is the maximum diameter of the network. This algorithm is also easily extended to the case of calls of limited duration, at the expense of an additive $O(\log D)$ factor in the competitive ratio, where $D$ is the ratio between the maximum and the minimum duration.

Awerbuch, Azar and Plotkin [AAP93] also prove lower bounds for the throughput version of the problem. A matching randomized lower bound $\Omega(nPBD)$ on a line network shows that the proposed algorithm is optimal up to a constant factor with respect to the various parameters of the problem, when condition (1) is satisfied.

On the contrary, if the transmission rate of a call can be up to a fraction $\frac{1}{k}$ of the bandwidth of a link, the authors show that any deterministic on-line algorithm has a very poor competitive ratio, an $\Omega(n^{1/k} + B^{1/k} + D^{1/k}$ lower bound is proved for this case on a line network of $n$ vertices. This lower bound holds for strictly competitive algorithms, for which we do not allow an additive constant factor. However, the competitive ratio is still $\Omega(\sqrt{n})$ if an additive constant factor is allowed.

**Probabilistic analysis and experimental results.** Kamath, Palmon and Plotkin [KPP96] study the virtual circuit routing problem in an intermediate model between worst case competitive analysis and probabilistic assumption on the network traffic. The requests arrive following a Poisson distribution while call durations are exponentially distributed. However the matrix of the transmission rates requested between any pair of vertices of the network is unknown and chosen by the adversary. Let $r$ be the maximum ratio between a transmission rate and the capacity of an edge, and let $\epsilon = \sqrt{r \log n}$. For the load balancing version, the author propose algorithms that achieve a worst case ratio, over all the traffic matrices, between the expected algorithm's congestion and the expected optimal congestion bounded by $1 + O(\epsilon)$. For the throughput version, the authors prove a worst case rejection ratio of $R^* + \epsilon$, where $R^*$ is the optimal expected rejection ratio. Observe that the performance of the algorithm improves if $r$ is smaller, a behaviour that is not observed in the AAP algorithm.

Algorithms based on these ideas have been implemented and tested on commercial networks [GKPR94]. A less conservative approach has been followed for these implementations. For instance, a smaller value of $\mu$ must be used for the exponential cost. The results obtained from the authors suggest that these strategies can behave better than greedy or reservation-based strategies in many practical scenarios.

**Multicast Routing.** On-line routing problems have been studied beyond point-to-point connection. Awerbuch and Azar [AA95] first considered multicast routing. Several multicast groups are active in a network. A tree connects all the nodes registered in a single group to a source node that transmits information to all the members of the group. At each step, one of the nodes of the network asks to join one of the group. The goal is to update the tree while minimizing the maximum load on a link of the network. The authors present an $O(\log n \log d)$ competitive algorithm, where d is the maximum number of nodes of the network that join a single group.

The throughput version of the multicast routing problem has first been addressed by Awerbuch and Singh [AS97]. In the throughput version the goal is to maximize the number of accepted subscriptions to a set of multicast groups that are active in the network. To accept a new subscription it is required to select a path that connects the node of the network to a spanning tree rooted at the source of the multicast group. This problem is qualitative different from the simple point-to-point connection where the algorithm knows the benefit obtained from a single connection when the corresponding circuit is assigned. In the multicast case, the investment of deploying a multicast tree in a given area of the network may be compensated only in the future, when a number of subscriptions from nodes of that area will be issued. It is fairly easy to see that a deterministic algorithm cannot obtain good performances even though the bandwidth required by each connection is small when compared with link capacity.

The proposed algorithm combines the approach based on assigning exponential costs to the network links, with a randomized algorithm presented by Awerbuch, Azar, Fiat and Leighton [AAFL96] for the following problem. A set of items are presented on-line. Each item belongs to one of different sets. The algorithm achieves a benefit only once it commits to one of the set. The benefit of the algorithm is the number of items in the selected set presented after the commitment. This problem arises in a set of application also motivated by network resources assignment. The authors present an algorithm with competitive ratio polylogarithmic in the number of items and in the number of sets. The algorithm derived in this way for the multicast problem also achieves a polylogarithmic competitive ratio in the number of multicast groups and in the size of the network. This algorithm is restricted to input sequences where the subscriptions to different multicast groups are not interleaved. The algorithm is involved at each time in the building only of a single multicast tree. The extended problem is furtherly studied by Goel, Rauch Henzinger and Plotkin [GHP], that show an algorithm that can simultaneously build a tree for all the multicast groups active in the network, while maintaining a polylogarithmic competitive ratio for the problem.

**Further applications of the exponential function.** Algorithms based on the idea of associating to each distinct resource of the network a cost that is exponential in the fraction of the resource currently assigned, are designed for various network problems other than virtual circuit routing.

The algorithms described in this section for the on-line virtual circuit routing problem are centralized algorithm. Awerbuch and Azar [AA94] propose admission control and route selection distributed algorithms, and evaluate the quality of the solution and the number of steps required to compute it. They show how a distributed algorithm can achieve a logarithmic competitive ratio for the load balancing version of the on-line virtual circuit routing problem. They also study the flow control problem, where each pair is presented together a fixed path that connects the two endpoints. The goal is to decide how much traffic to route on each path connecting a pair, with the goal of maximizing the overall flow in the network.

An on-line variant of packet routing is considered by Awerbuch, Azar and Fiat [AAF96a]. Each packet injected in the network is specified by a source node, a destination node and a time of release. There are buffers of limited capacity at the nodes, while we assume that a packet takes one unit of time to cross a link. The goal is to minimize the delay of packets to reach their destination, while keeping limited the load on the links and the number of packets in the buffers. The authors present an algorithm that is $O(1)$ competitive with respect to the average and the maximum delay, while the capacity of the links and of the buffers is increased by at most an $O(\log nT)$ factor, where $T$ is the maximum delay of the off-line schedule.

Leonardi and Marchetti-Spaccamela [LMS] show how a general algorithm based on the use of the exponential function solves an on-line version of positive linear programming, a model suitable for a large number of routing and scheduling problems in their load balancing and throughput version. A constraint is associated with each distinct resource, for instance the use of a link for one unit of time, and each of the requests can be satisfied if one among a set of combinations of the available resources is assigned to the request. The proposed algorithm is $O(\log m)$ competitive, where $m$ is the number of distinct resources in the system.

## 2.2 The edge-disjoint paths problem

We have described in the previous section efficient algorithms for the virtual circuit routing problem for any network topology for its load balancing version, and the throughput version restricted to the case of small bandwidth call request. In this section we describe algorithms for the case of calls with bandwidth request up to the whole capacity of a link. In this case, a polynomial lower bound on the competitive ratio of deterministic on-line algorithms is proved [AAP93] also for a line network topology.

A good effort has been devoted to the study of *randomized* on-line algorithms for network topologies as line, trees and meshes. (Deterministic algorithms with logarithmic competitive ratio are however still possible for specific topologies like expander graphs [KR96]).

The throughput version of the on-line virtual circuit routing problem contains as a special case the on-line version of the edge-disjoint paths problem. A general technique allows to reduce the problem on a network with edges of equal capacity to the on-line edge disjoint paths problem (a similar technique is not known if the edges have different capacities.) Randomized logarithmically competitive on-line algorithms have been devised for a line network [GGK+93], trees [ABFR94, AGLR94, LMSPR98], meshes [AGLR94, KT95], and for "densely embedded, nearly Eulerian graphs" [KT95]. However the question of the existence of an efficient randomized algorithm for any network topology has a negative answer, since a polynomial lower bound can be proved for a specific topology [BFL96].

We will describe these results in the following of this section.

**From virtual circuit routing to edge-disjoint paths.** Awerbuch, Bartal, Fiat and Rosèn [ABFR94] show how the design of randomized algorithms for the problem on a network with edges of uniform capacities can be reduced to the design of randomized algorithms for the edge-disjoint paths problem. The parameters involved in the problem are the benefit, the duration and the bandwidth. The authors show how to deal with each one of these parameters at the expenses of a factor $O(\log \Delta)$ in the competitive ratio, where $\Delta$ is the ratio between the maximum and the minimum value of the parameter.

This result is obtained using a "classify and randomly select" technique. Assume that the individual benefit of a call is ranging between 1 and $B$, and that a $c$ competitive algorithm is available if calls have equal benefits. We partition the interval $[1, B]$ into intervals $[2^i, 2^{i+1}]$, $i = 0, \ldots, \lceil \log B \rceil - 1$. A call is said of class $i$ if its benefit is in the $i$th interval. The algorithm chooses an integer uniformly at random in the set $\{0, \ldots \lceil \log B \rceil - 1\}$. A $c$ competitive algorithm for calls of equal benefit is then applied to calls of class $i$ when presented, while calls of other classes are rejected.

An $O(c \log B)$ competitive ratio for the whole algorithm can be shown as follows. Denote with $ON_i$ and with $OPT_i$ the benefit of the on-line and of the optimal algorithm when restricted to deal only with calls of the $i$th class. If the algorithm selects the $i$th class, then its benefit is at least $\frac{1}{2c}OPT_i$, since the benefits of calls of class $i$ differ for at most a factor of 2. We then have for the overall algorithm:

$$
\begin{aligned}
ON &\geq \sum_{i=0}^{\lceil \log B \rceil - 1} E(ON_i) \\
&\geq \frac{1}{2c} \frac{1}{\lceil \log B \rceil} \sum_{i=0}^{\lceil \log B \rceil - 1} OPT_i \\
&\geq \frac{1}{2c \lceil \log B \rceil} OPT
\end{aligned}
$$

A similar technique also allows to deal with varying duration and bandwidth requests. Lipton and Tomkins [LT94] devise a technique suitable for the case of parameters whose range of variation is not known at the beginning of the execution of the algorithm. This technique obtains an $O(\log^{1+\epsilon} \Delta)$ competitive ratio, with $\epsilon > 0$ arbitrarily small, where $\Delta$ is the range of variation of the parameter observed during the whole sequence.

**Edge-disjoint paths on trees.** The problem for trees has first been addressed by Awerbuch, Bartal, Fiat and Rosèn [ABFR94]. They propose an $O(\log n)$ competitive algorithm also based on a classify a randomly select technique.

All the vertices of the tree are partitioned into $O(\log n)$ different classes, on the basis of a recursive application of a balanced tree separator.

A balanced tree separator [vLe90] is a vertex whose removal disconnect the tree into pieces of at most $\frac{2}{3}n$ vertices. The application of the separator to the initial tree $T = T_0$ returns a vertex that is assigned with level 0. The tree is then splitted into a set of subtrees of level 1, the generic one we denote with $T_1$. All the separators of the trees of level $i$ are assigned with level $i$. Their removal produce a set of trees of level $i + 1$. The process is recursively iterated for a logarithmic depth until we obtain fragments formed by single vertices.

Calls are then partitioned in $O(\log n)$ classes. A call is assigned to class $i$ if the vertex of lower level included in the path connecting the two endpoints has level $i$. One of the classes is chosen at random at the beginning, and the algorithm restricts to deal with calls of that class using a greedy strategy. The greedy strategy accepts a call if it does not intersect any previously accepted call. When restricted to calls of a class $i$ , the greedy algorithm is 2 competitive since a call of class $i$ includes at most two edges that are adjacent to a vertex of level $i$, and then can block at most 2 calls of level $i$ presented in the future that can be accepted together in a solution. Following the scheme of analysis presented in the previous paragraph, the algorithm can be easily proved to be $O(\log n)$ competitive.

The asymptotic competitive ratio of this algorithm has been improved to $O(\log D)$ where $D$ is the diameter of the tree, from Awerbuch, Gawlick, Rabani and Leighton [AGLR94]. A matching $\Omega(\log D)$ randomized lower bound is proved for trees of diameter $D$ [ABFR94, AGLR94].

Both algorithms have an asymptotically good competitive ratio, but suffer the drawback to achieve a good solution only with small probability, while the main contribution to the average is given by a very high benefit achieved with low probability. An $O(\log D)$-competitive algorithm that achieves on any sequence a benefit close to the expectation with good probability is proposed by Leonardi, Marchetti-Spaccamela, Presciutti and Rosèn [LMSPR98].

In a first step, an on-line deterministic filter is applied to the input sequence. Each call that is not filtered out is called a *candidate call*. The authors prove the existence of an on-line deterministic filter that produces a set of candidate calls $\mathcal{C}$ whose number is at least $1/4$ of the optimal solution. Moreover, each candidate call is shown to intersects at most $O(\log D)$ previously candidate calls. (Such filter can also be based on the

AAP algorithm when the on-line algorithm is assumed to have logarithmic more capacity than the optimal solution.) The set $\mathcal{C}$ forms a new input sequence to an on-line randomized selection procedure that declares each candidate "to be considered" with probability $p = O(\frac{1}{\log D})$, and accepts a considered candidate if not intersecting any previous considered candidate, otherwise reject.

For an appropriate value of $p$, the algorithm is shown to have $O(\log D)$ competitive ratio and achieve any constant fraction of the benefit with at least constant probability. For a different tuning of $p$, the algorithm has a competitive ratio of $O(\log^{1+\epsilon} D)$, with arbitrarily small $\epsilon > 0$, but achieves any constant fraction of the expected solution with probability that tends asymptotically to 1.

**Edge-disjoint paths on meshes.** An $O(\log n)$ competitive algorithm for meshes is proposed by Kleinberg and Tardos. Awerbuch et al. [AGLR94] gave a previous $O(\log n \log\log n)$ algorithm and an $\Omega(\log n)$ randomized lower bound.

The algorithm proposed by Kleinberg and Tardos, KT in the following, partitions the $n \times n$ mesh into $\frac{n^2}{\log^2 n}$ squares of size $\log n \times \log n$. Calls that have both endpoints in the same square are *short calls*, calls that have the two endpoints in different squares are *long calls*. KT first tosses a coin and decides to accept only short calls, or only *long calls*.

The basic idea of KT is to deal with long calls using an auxiliary network, called "simulate network", with logarithmic capacity on the edges, and running a version of the AAP algorithm on this network.

The simulate network is constructed associating a vertex with each submesh, and connecting with edges vertices associated with adjacent submeshes. The logarithmic capacity on the edges is to model that at most a logarithmic number of edge-disjoint paths can be routed through adjacent meshes.

A part of the mesh selected at random is then dedicated to a crossbar structure used to route long calls.

Short calls are routed using a greedy algorithm that is easily proved to be $O(\log n)$ competitive.

Long calls are rejected if one of the endpoints are within the crossbar structure because all the paths leading to this endpoint can be easily blocked from other calls routed in that region. A long call is also rejected if a previous long call with an endpoint in the same submesh has has been accepted. This is possible still keeping a logarithmic competitive ratio since at most a logarithmic number of long calls with an endpoint in a given submesh can be accepted in a solution. If not, the call in the original mesh is transformed into a call in the simulate network between the two vertices associated with the two submeshes that contain the two endpoints of the call.

The sequence of calls in the simulate network so formed is the input of the $O(\log n)$ competitive AAP algorithm. The algorithm returns a path in the simulate network for those calls that are accepted. The path in the simulate network indicates a sequence of submeshes to cross to connect the two endpoints. This path is then transformed into a path in the original mesh that is edge-disjoint with all the previous accepted calls.

Kleinberg and Tardos extends the idea of this algorithm to obtain an $O(\log n)$ competitive algorithm also for "densely embedded, nearly Eulerian" planar graphs [KT95].

This algorithm has the drawback to obtain a benefit 0 with probability 1/2 if all the calls are either short or long. Leonardi, Marchetti-Spaccamela, Presciutti e Rosèn [LMSPR98] give a variant of the Kleinberg-Tardos algorithm that obtains any constant fraction of the expected solution with probability that tends to 1 as the optimal solution is asymptotically bigger than $\Theta(\log^4 n)$.

**A lower bound on randomized algorithms for arbitrary networks.** The obvious question was if there exists an efficient, polylogarithmic, randomized competitive algorithm for any network topology. Bartal, Fiat and Leonardi [BFL96] proved an $\Omega(n^\epsilon)$ lower bound on on-line edge-disjoint paths for a specific network.

An $\Omega(n^\epsilon)$ lower bound is first devised for an on-line version of the independent set problem. A graph $G$ from which the vertices of the sequence are drawn is known since the beginning to the algorithm. Every time a new vertex is presented, the algorithm can accept the vertex only if it is not adjacent to any previously accepted vertex.

The lower bound for the on-line independent set problem is then transformed into a lower bound for the on-line edge-disjoint path problem. This is done by embedding the graph and the input sequence for the independent set lower bound into a set of calls on a specific network. Each vertex of the graph is mapped to a call of the network. The embedding satisfies the property that the vertices of any independent set of the graph are mapped to a set of calls that can be connected with mutually edge-disjoint paths, while two adjacent vertices are mapped to two calls that can only be connected with intersecting paths. An $\Omega(n^\epsilon)$, with $\epsilon = 2/3(1 - \log_4 3)$ is devised in this way for the problem.

### 2.3 Other models

In the previous sections we have described on-line virtual circuit routing algorithms where connections request are accepted or rejected when presented, and the obtained benefit is either arbitrary or equal for all the communication requests.

In this section we describe the results obtained for different models. I.e., calls can be preempted and/or rerouted at some later time after their acceptance, or the benefit accrued from a call can be proportional to the amount of resources assigned to this call.

The competitiveness of on-line call control algorithms has actually been first studied by Garay and Gopal [GG92] when call preemption is allowed and the benefit of a call is its holding time. Calls that are accepted can be later preempted, while calls that are rejected cannot be resumed in the future. Call preemption may allow the on-line algorithm to supply part of the lack of knowledge of the future. The authors prove that if the holding time of a call is unknown at the time the call request is presented, no competitive algorithm can exist for this model. On the other hand, if the benefit obtained from a call until its preemption is not lost, an on-line algorithm is able to perform as well as the optimal solution.

In many practical scenarios this is not the case, if a call is interrupted, the corresponding gain is lost, or the connection is started again from the beginning. Garay, Gopal, Kutten, Mansour and Yung [GGK+93], consider this problem on a line network for different benefit functions. If the benefit obtained from calls that are terminated is the number of links of the route, the so called telephone model, then they present an algorithm with constant competitive ratio equal to $\frac{1}{2g+1}$ where $g$ is the golden ratio. This is obtained by the following simple idea: the algorithm accepts a new call if its benefit is at least $g$ times the benefit lost for preempting calls. If the benefit of a call is the holding time of the call, then a competitive ratio of $1/4$ is achieved. If the benefit is constant for each call, independently from the length of the link, the competitive ratio is only logarithmic in the number of vertices of the network. They also prove that no on-line deterministic algorithm can achieve a better ratio.

Further work for preemptive algorithms has been done by Bar-Noy, Canetti, Kutten, Mansour and Schieber [BNCK+95] assuming that the benefit obtained from calls that are terminated is the product bandwidth-holding time. They show how the best competitive ratio of an algorithm is parameterized by the maximum fraction $\delta$ of the bandwidth of a link that is requested by a single call. They first study the problem on a single link and present deterministic algorithms with almost optimal constant competitive ratio as soon as $\delta < 1$, while for $\delta = 1$ they prove that deterministic algorithms cannot achieve a bounded competitive ratio. Similar results are obtained on a line network if $\delta \leq 1/2$ and calls have constant duration. Canetti and Irani [CI95] prove a randomized logarithmic preemptive lower bound even for a single link, if calls have arbitrary durations and can potentially obtain the whole bandwidth ($\delta = 1$). Whether randomized preemptive algorithms for a line network with calls of constant benefit and infinite duration can achieve a competitive ratio better than a logarithmic factor is still an open problem.

Bartal, Fiat and Leonardi [BFL96] prove an $\Omega(n^\epsilon)$ preemptive randomized lower bound for arbitrary networks when the benefit of each call is constant, thus showing that it does not exist a polylogarithmic call control algorithm for any network even if preemption and randomization are allowed.

Finally, we mention the possibility to reroute virtual circuits. This has been proved useful when one seeks for minimizing the load in a network generated by calls with unknown holding times. Awerbuch, Azar, Plotkin and Waarts [AAPW94] prove that if an $O(\log n)$ number of rerouting is allowed, it is possible to design an $O(\log n)$ competitive algorithm for load balancing in arbitrary topology network.

# 3 On-line routing in optical networks

On-line wavelength routing problems in all-optical networks have motivated several works in the area of on-line routing. All-optical communication networks exploit the optical technology for both switching and communications functions.

Information is transmitted as light rays from source to destination without any electronic conversion in between. The Wavelength Division Multiplexing technology (WDM) consists in partitioning the available bandwidth on each link in many channels, each at a different optical wavelength. This allows the parallel transmission on an optic fiber link of different data streams, with speed related to the assigned wavelength.

The main constraint the wavelength allocation must obey is that for a link, on a given wavelength, only one signal can be transmitted. Two data streams transmitted on a link must be assigned with different wavelengths. Conversion of data between different wavelengths is limited by the current technology If we assume that wavelength conversion is not allowed, a data stream is then assigned to a single wavelength between the transmitter and the receiver.

In general a WDM network consists of routing photonic switches connected through point-to-point optic fiber links. In the following we will mainly consider *switched* optical networks. In these networks, routing switched are able to direct any pair of incoming data streams carried on the same wavelength to any pair of outgoing links where such wavelength is available.

In contrast, *switchless* optical networks have been considered. In this model, the routing pattern at each switch is fixed for each wavelength, meaning that each data stream entering the routing node on a given wavelength is directed to a predetermined fixed subset of the outgoing edges on which the wavelength is available. As a result, a wavelength cannot be used on all the links reached by the propagation of the light ray of an ongoing communication.

Unless specified otherwise, we will refer in the following to the switched model.

**The model.** We model an optical network as a graph, whose vertices are switching routers with possible connected terminals, and links are optic fibers. Edges representing optic fibers should be directed in one single direction as a matter of fact that optic amplifiers are directed devices. This is reflected by modeling each optical link with two directed links in opposite direction rather than a single undirected link. However, many of the results presented for the undirected model, extend under certain restrictions (usually at the expenses of a constant factor in the competitive ratio) to the directed bidirectional model. Finally, the same set of wavelengths is considered potentially available on all the links of the network.

Each communication request in a sequence specifies a pair of vertices $(s, t)$. An algorithm for wavelength routing in general deals with two kind of problems: *Wavelength selection* and *Routing assignment*. The wavelength selection problem consists in selecting a wavelength $w$ for the communication. The routing assignment problem consists of choosing a path connecting node $s$ to $t$ such that each link of the path is not assigned to any other communication on wavelength $w$.

The routing problem on WDM switched networks has been often referred to as *path coloring*. Given a graph $G = (V, E)$ representing the network, we are given a sequence $(s_i, t_i)$ of requests consisting of pairs of vertices of the graph $G$. The algorithm must assign to each pair $(s_i, t_i)$ a color (wavelength), and a path connecting $s_i$ to $t_i$ such that two paths sharing an edge are not associated with pairs of same color.

As for the *Virtual circuit routing* problem, a *load* version and a *throughput* version have been studied.

In the load version, all the communication requests must be accepted with the goal of minimizing the overall number of colors used.

In the throughput version, the number of available colors is considered fixed, while the goal is that of maximizing the number of communications that can be accepted given the network topology and the number of available wavelengths.

Although path coloring is at the basis of the study of optical routing, a number of additional features can be added to the problem. Communications can have limited durations, or, in the throughput version, varying benefits. We will not discuss further this second problem, that can be for instance dealt, as in the virtual circuit routing problem, using a Classify and Randomly Select technique [ABFR94].

## 3.1 The load version of path coloring

In the load version of the path coloring problem, all the communication requests must be accepted. A color and a path must be assigned to each call, with the goal of minimizing the number of used colors.

A natural question is the relation between the path coloring problem and the load balancing problem in networks of Section 2.1, where we seek to minimize the maximum number of paths crossing an edge. Given an instance for path coloring, e.g. a sequence of pairs of vertices in a network, the optimal solution for the corresponding load balancing problem is a clear lower bound on the optimal number of colors required for the path coloring problem. The size of the optimal load and of the optimal number of colors are actually equal for those topologies where a pair of vertices is connected by one single path (e.g. a line or a tree), while there are examples of networks where they are far away for a factor $O(n)$ on certain instances.

However, all the algorithms presented in the following for various network topologies are shown to be competitive against an adversary for which the optimal load is used as a lower bound on the optimal solution. For this reason, we will denote this measure by $\Lambda^*$ throughout this section.

The on-line path coloring problem has been studied for the following network topologies: line, rings, trees, meshes. Efficient competitive algorithms have been devised for all these network topologies.

In contrast, an $\Omega(n^\epsilon)$ lower bound on the competitive ratio of randomized algorithms working for arbitrary network topologies has been shown by Bartal, Fiat and Leonardi [BFL96]. This lower bound is achieved in a way similar to the lower bound for the on-line edge-disjoint paths problem (see Section 2.2). A lower bound on the the on-line graph coloring problem is first established and then it is turned into a lower bound for on-line path coloring in a network shaped as a brick wall.

**Path coloring on the line.** It turns out the the on-line path coloring problem on an undirected line network has been studied even before the notion of competitive analysis has been formally introduced. This problem is equal to the *On-line interval graph coloring* problem.

Each vertex of an interval graph is mapped to an interval on the line. Two vertices of the graph are adjacent if and only if the corresponding intervals are overlapping. Since an interval graph is a perfect graph, the optimal solution uses exactly $\Lambda^*$ colors, and can be found in polynomial time for example with a simple divide and conquer technique.

The problem of coloring on-line an interval graph has been first studied by Kierstead and Trotter [KT81]. They present an optimal on-line deterministic algorithm whose competitive ratio is at most $3\Lambda^* - 2$ and prove that no deterministic algorithm can achieve a better ratio. These results immediately extend to path coloring on a line. More details on on-line interval graph coloring can be found in Chapter **??**.

**Path coloring on rings.** In the wavelength routing problem on rings a communication between a pair of vertices can be routed with one of the two possible paths connecting the two vertices. A simple approach to the problem used by several authors (for instance [GK97] and [MKR95]) consists of cutting the ring at any edge and then transforming the problem on a line network. The number of paths that cross this edge in an optimal solution is at most equal to the optimal load. Then, the load on any other edge in the resulting problem on a line network is increased by at most $\Lambda^*$ and then the optimal load is at most doubled. An algorithm for the line can now be applied yielding an algorithm for the ring whose competitive ratio is away from the competitive ratio of the algorithm for the line by at most a multiplicative factor of 2.

If each pair can be connected with only one of the two possible paths, then the problem reduces to the *circular arc graph coloring* problem. In a circular arc graph, each vertex can be mapped on on an arc of a ring, and two vertices are connected with an edge if and only if the two arcs are overlapping. For this problem, Ślusarek has shown that an on-line algorithm can use a number of colors bounded by $3\Lambda^* - 2$ then yielding the same result for interval graph coloring.

**Path coloring on trees.** An $O(\log n)$-competitive algorithm for trees has been proposed by several authors [BL97, BKS96, GSR96]. The problem of on-line path coloring on a tree can be reduced to the problem of coloring on-line an $O(\Lambda^*)$-inductive graph. A graph is $d$-inductive if the vertices can be associated with numbers 1 through $n$ in a way that each vertex is connected to at most $d$ vertices with higher numbers. Given an instance of the path coloring problem on a tree, the so called *intersection graph* is built by associating a vertex with each path, and connecting two vertices with a link if the two corresponding paths are intersecting. The intersection graph can be shown to be $2(\Lambda^* - 1)$-inductive [BL97]. An algorithm by Irani [Ira94] that shows how to color on-line a $d$ inductive graph with $O(d \log n)$ colors can then be applied to yield an $O(\log n)$ competitive algorithm.

An almost matching deterministic $\Omega(\frac{\log n}{\log \log n})$ lower bound has also been proved by Bartal and Leonardi [BL97]. This lower bound has been obtained on a tree of depth $\log n$. This shows that an $O(\log \Delta)$ lower bound, where $\Delta$ is the diameter of tree, is not possible at least for deterministic algorithms. (In contrast, recall that an $O(\log \Delta)$ randomized competitive algorithm for edge disjoint paths on trees is possible).

**Wavelength assignment with limited duration.** A natural variation of the problem consists in considering connection requests of limited duration. Pairs of vertices arrive and depart in an on-line fashion. A color must be assigned to the pair when the call arrives, the color can be reused after that the call is ended. An algorithm with logarithmic competitive ratio has been devised for this problem for line, tree and ring topologies from Gerstel, Sasaki and Ramaswami [GSR96].

The algorithm for line and tree topologies uses a balanced separator-based idea similar to that for the $O(\log n)$ algorithm for call control on trees [ABFR94]. In a preprocessing phase the graph is recursively separated through a balanced edge separator. The tree is broken in many pieces and a specific set of colors is reserved to those paths crossing the separator. A similar procedure is then applied recursively for a depth at most logarithmic until we end with trees formed by single edges.

Let $\Lambda^*$ be the maximum number of pairs crossing a given edge that are active at the same time. One can prove that $\Lambda^*$ colors for each level of separation are enough to color all the calls at their arrival. $\Lambda^*$ is also an obvious lower bound on the optimal solution, then yielding the $O(\log n)$ competitive algorithm. The algorithm can also be extended to rings using again the trick of cutting the ring at any link.

**Path coloring on meshes.** The problem of path coloring on meshes has been approached by Bartal and Leonardi [BL97] using the idea of partitioning the meshes in submeshes of logarithmic size. This follows the solution proposed by Kleinberg and Tardos for the on-line edge-disjoint paths problem on meshes [KT95]. However, in this case all the calls must be assigned with a color, while in the edge-disjoint paths problem rejection is allowed.

Calls are divided between *short calls*, with both endpoints in the same submesh, and *long calls*, with endpoints in different submeshes. Disjoint set of colors are dedicated to long calls and short calls.

Short calls are routed through a shortest path connecting the two vertices. Short calls in different submeshes are then non-conflicting, and the colors can be reused in different submeshes. The colors are assigned to short calls in a greedy manner according to their relative length. This first part of the algorithm yields an $O(\log n)$ competitive ratio with respect to an optimal algorithm for short calls.

The part of the algorithm that deals with long calls transforms the problem into a *path coloring problem with more bandwidth* in a *simulated network* where each fiber optic connection between two vertices is formed by a link with logarithmic capacity. This results in the fact that a logarithmic number of paths with same color can include a link.

A vertex of the simulated network is associated with each submesh, while two vertices are connected by an edge with logarithmic capacity if the two corresponding submeshes are adjacent. This is to model that at most a logarithmic number of paths with same color can be routed through two adjacent submeshes.

An $O(\log n)$-competitive algorithm for the path coloring problem with logarithmic bandwidth is then obtained. This algorithm uses an exponential function as in the algorithm for on-line load balancing (see Section 2.1. The proposed algorithm is an $O(\log n)$ competitive algorithm on general networks if a logarithmic bandwidth is allowed on any color.

The instance in the original into an instance in the simulated network by replacing each call in the original mesh with a call between the two vertices in the simulated network that represent the two submeshes containing the two vertices of the call. Each call in the simulated network is then given to the algorithm for path coloring with more bandwidth, that assigns a color and a route in the simulated network. This route is then converted on-line into a route in the original mesh preserving the property that paths with same color are edge-disjoint.

The part of the algorithm dealing with long calls has also a logarithmic competitive ratio then yielding together with the algorithm for short calls an $O(\log n)$ competitive algorithm for the problem.

The lower bound on the competitive ratio of randomized on-line algorithms for the on-line load balancing problem for virtual circuit routing [BL97] also implies that the algorithm for path coloring on meshes is optimal up to a constant factor.

### 3.2 The throughput version of path coloring

The throughput version of path coloring consists of maximizing the number of communication requests that can be accepted for a given set of wavelengths and a given network topology. Let us assume that for a given network topology, a competitive algorithm is available for the on-line edge-disjoint path problem described in Section 2.2. This can also be considered an algorithm for the throughput version of the path coloring problem when one single wavelength is available. A simple technique proposed by Awerbuch, Azar, Fiat, Leonardi and Rosèn [AAF+96b] allows to transform an algorithm for a single wavelength into an algorithm for the multi wavelength case at the expenses of an additive term of 1 in the competitive ratio.

The algorithm uses a first fit approach. Each one of the $\Lambda$ colors are assigned with a number from 1 to $\Lambda$. A different copy of the algorithm for on-line virtual circuit routing is then executed for each one of the colors. Every time a new call is presented we apply a first fit based approach. The call is given as an input to the algorithm for the first color. If this algorithm accepts the call, than it is assigned with the first color and the route chosen by the algorithm. If the algorithm for the first color rejects the call, this is given as input to the algorithm for the second color, and so on until the call is eventually accepted by an algorithm for a color, or rejected by all the colors. In this last case the call is rejected by the overall algorithm.

Let $c$ be the competitive ratio of the base algorithm (deterministic or randomized) for virtual circuit routing. The algorithm described above for the throughput version of on-line path coloring achieves a competitive ratio of at most $c + 1$. This allows to state the existence of an $O(\log n)$ competitive algorithm for those topologies for which there exists a competitive $O(\log n)$ competitive algorithm for on-line virtual circuit routing.

If we look to algorithms for general networks, an $\Omega(n^{\epsilon})$ lower bound immediately follows from the analogous lower bound for the on-line virtual circuit routing problem [BFL96].

The throughput version has also been studied for other models of optical networks, namely switchless optical networks [AAF+96b]. An $O(\log n)$ randomized competitive algorithm for trees has been obtained by devising an $O(\log n)$-competitive algorithm for the single wavelength case and then extending it to deal with an arbitrary number of wavelengths.

## References

[AA94] B. Awerbuch and Y. Azar. Local optimization of global objectives: competitive distributed deadlock resolution and resource allocation. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 240–249, 1994.

[AA95] B. Awerbuch and Y. Azar. Competitive multicast routing. *Wireless Networks*, 1:107–114, 1995.

[AAF+93] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line load balancing with applications to machine scheduling and virtual circuit routing. In *Proceedings of the 25th ACM Symposium on the Theory of Computing*, pages 623–631, 1993.

[AAF96a] B. Awerbuch, Y. Azar, and A. Fiat. Packet routing via min-cost circuit routing. In *Proceedings of the 4th Israeli Symposium on Theory of Computing and Systems*, pages 37–42, 1996.

[AAF+96b] B. Awerbuch, Y. Azar, A. Fiat, S. Leonardi, and A. Rosén. On-line competitive algorithms for call admission in optical networks. In *Proceedings of the 4th Annual European Symposium on Algorithms, Lecture Notes in Compter Science 1136*, pages 431–444. Springer-Verlag, 1996.

[AAFL96] B. Awerbuch, Y. Azar, A. Fiat, and T. Leighton. Making commitments in the face of uncertainty: How to pick a winner almost every time. In *Proc. of the 28th Annual ACM Symposium on Theory of Computing*, pages 519–530, 1996.

[AAP93] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive online routing. In *34th IEEE Symposium on Foundations of Computer Science*, pages 32–40, 1993.

[AAPW94] B. Awerbuch, Y. Azar, S. Plotkin, and O. Waarts. Competitive routing of virtual circuits with unknown duration. In *Proceedings of the 5th ACM-SIAM Symposium on Discrete Algorithms*, pages 321–327, 1994.

[ABFR94] B. Awerbuch, Y. Bartal, A. Fiat, and A. Rosén. Competitive non-preemptive call control. In *Proc. of 5th ACM-SIAM Symposium on Discrete Algorithms*, pages 312–320, 1994.

[ABK92] Y. Azar, A. Broder, and A. Karlin. On-line load balancing. In *Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science*, pages 218–225, 1992.

[AGLR94] B. Awerbuch, R. Gawlick, T. Leighton, and Y. Rabani. On-line admission control and circuit routing for high performance computing and communication. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 412–423, 1994.

[AKP+93] Y. Azar, B. Kalyanasundaram, S. Plotkin, K. Pruhs, and O. Waarts. Online load balancing of temporary tasks. In *Proceedings of the 3rd Workshop on Algorithms and Data Structures*, LNCS. Springer-Verlag, 1993.

[AS97] B. Awerbuch and T. Singh. On-line algorithms for selective multicast and maximal dense trees. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 354–362, 1997.

[BFL96] Y. Bartal, A. Fiat, and S. Leonardi. Lower bounds for on-line graph problems with application to on-line circuit and optical routing. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 531–540, 1996.

[BKS96] A. Borodin, J. Kleinberg, and M. Sudan, 1996. Personal communication.

[BL97] Y. Bartal and S. Leonardi. On-line routing in all-optical networks. In *Proceedings of the 24th International Colloqium on Automata, Languages and Programming*, LNCS 1256, pages 516–526. Springer-Verlag, 1997.

[BNCK+95] A. Bar-Noy, R. Canetti, S. Kutten, Y. Mansour, and B. Schieber. Bandwidth allocation with preemption. In *Proc. of the 27th Symposium on Theory of Computation*, pages 616–625, 1995.

[CI95] R. Canetti and S. Irani. Bounding the power of preemption in randomized scheduling. In *Proceedings of the 27th ACM Symposium on Theory of Computing*, pages 616–615, 1995.

[Fra] A. Frank. Packing paths, cuts, and circuits – a survey. In H.J. Pr B. Korte, L. Lovász, editor, *Paths, Flows, and VLSI-Layout*.

[GG92] J.A. Garay and I.S. Gopal. Call preemption in communications networks. In *Proceedings of INFOCOM '92*, pages 1043–1050, 1992.

[GGK+93] J.A. Garay, I.S. Gopal, S. Kutten, Y. Mansour, and M. Yung. Efficient on-line call control algorithms. In *Proceedings of the 2nd Israeli Symposium on Theory of Computing and Systems*, pages 285–293, 1993.

[GHP] A. Goel, M. Rauch Henzinger, and S. Plotkin. Online throughput-competitive algorithm for multicast routing and admission control. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete algorithms*.

[GK97] O. Gerstel and S. Kutten. Dynamic wavelength allocation in WDM ring networks. In *Proc. of ICC '97*, 1997.

[GKPR94] R. Gawlick, A. Kamath, S. Plotkin, and K. Ramakrishnan. Routing and admission control of virtual circuits in general topology networks. Technical Report BL011212-940819-19TM, AT&T Bell Laboratories, 1994.

[Gre92] P.E. Green. *Fiber-Optic Communication Networks*. Prentice Hall, 1992.

[GSR96] O. Gerstel, G.H. Sasaki, and R. Ramaswami. Dynamic channel assignment for WDM optical networks with little or no wavelength conversion. In *Proceedings of the 34th Allerton Conference on Communication, Control, and Computin*, 1996.

[Ira94] S. Irani. Coloring inductive graphs on-line. *Algorithmica*, 11:53–72, 1994. Also in Proceedings of the 31st IEEE Symposium on Foundations of Computer Science, 1990, 470-479.

[KPP96] A. Kamath, O. Palmon, and S. Plotkin. Routing and admission control in general topology networks with poisson arrivals. In *Proceedings of the 7th ACM-SIAM Symposium on Discrete Algorithms*, pages 269–278, 1996.

[KR96] J. Kleinberg and R. Rubinfield. Short paths in expander graphs. In *Proceedings of the 37th Ann. IEEE Symposium on Foundations of Computer Science*, pages 86–95, 1996.

[KT81] H. A. Kierstead and W. T. Trotter. An extremal problem in recursive combinatorics. *Congressus Numerantium*, 33:143–153, 1981.

[KT95] J. Kleinberg and E. Tardos. Disjoint paths in densely embedded graphs. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 52–61, 1995.

[LMP+91] T. Leighton, F. Makedon, S. Plotkin, C. Stein, E. Tardos, and S. Tragoudas. Fast approximation algorithms for multicommodity flow problems. In *Proceedings of the 23rd ACM Symposium on Theory of Computing*, pages 101–111, 1991.

[LMS] S. Leonardi and A. Marchetti-Spaccamela. On-line resource management with applications to routing and scheduling. In *Proceedings of the 23rd International Colloqium on Automata, Languages and Programming*. Springer-Verlag.

[LMSPR98] S. Leonardi, A. Marchetti-Spaccamela, A. Presciutti, and A. Rosèn. On-line randomized call-control revisited. Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.

[LT94] R. J. Lipton and A. Tomkins. Online interval scheduling. In *Proc. of the 5th ACM-SIAM Symposium on Discrete Algorithms*, pages 302–311, 1994.

[MKR95] M. Mihail, C. Kaklamanis, and S. Rao. Efficient access to optical bandwidth. In *Proc. of the of 36th Annual IEEE Symposium on Foudations of Computer Science*, 1995.

[MP97] Y. Ma and S. Plotkin. Improved lower bounds for load balancing of tasks with unknown duration. *Information Processing Letter*, 62:31–34, 1997.

[vLe90] Jan van Leeuwen ed. *Handbook of theoretical computer science, Vol A, Algorithms and Complexity.* The MIT Press, 1990.