A Routing and Resource Preservation Strategy for Traffic Engineering in Communication Networks

Martin Heusse[a b] – Annie Gravey[b] * †

[a]LSR-IMAG BP 72, 38402 St Martin d'Hères cedex, France.
Martin.Heusse@imag.fr

[b]ENST Bretagne BP 832, 29285 Brest Cedex, France

This paper presents a method for dynamic load balancing in data networks. When multiple routes are available, it determines their load shares as a function of a composite metric that takes into account the paths' length and load. A general resource preservation mechanism is also presented that complements the proposed random routing strategy when the network is heavily loaded. We compare our approach with trunk reservation in the particular case of fully meshed networks and evaluate its performance in any network, where an equivalent mechanism is missing. We validate our approach by means of simulation and provide insights on the routing solutions that it obtains.

1. Introduction

This paper details a load sharing strategy that aims at providing a better network utilization than current routing approaches. The core of the present proposition is to send the load not only on the best path at the moment, but rather to spread it on all comparable paths with probabilities computed based on the quality of each path. The probabilities represent the relative quality of a path with respect to the others.

Although load balancing is generally recognized as desirable to use efficiently the communication equipments, it appears as a challenging task in many respects. First it has to be adapted to the network capabilities in terms of flow management. Protocols like MPLS [14] address this issue. They bring to packet switched networks management capabilities that were available only in connection oriented networks (or with ATM, for instance). Second, from the performance viewpoint, load balancing must be stable in the sense that the path set of each flow across the network should not vary significantly over short periods of time. Furthermore, network resources should be preserved so that the network keeps accepting flows possibly at the cost of the load balancing accuracy.

This paper describes a load balancing proposal that takes into account these objectives. For convenience, it is denominated hereafter as Random Shaped Routing (RSR). It is a dynamic routing strategy that can be used to enhance routing protocols like PNNI or applied to the routing of MPLS label switched paths (LSP). RSR is not based on a

*The authors would like to thank Nicolas Broutin and Yvon Kermarrec for their helpful comments.
†This work has been partially supported by France Telecom R&D.

centralized multi-commodity flow routing computation (see [2,3]) and consequently leads to less optimal routing patterns, especially for demands with bandwidth constraints of the same order of magnitude as the transport infrastructure. Rather, it is adaptive in the sense that it is able to track continuously the load variations. Although it stems from a different approach, RSR takes over in data networks the same task as DAR (Dynamic Alternative Routing [5]) does in telephone networks and we show that the resulting routing is comparable in this particular case.

The paper is organized as follows. Section 2 gives an overview of RSR. Section 3 gives insights on the routing obtained by RSR, comparing it with other strategies proposed for traffic engineering in data networks. Similarly to DAR, the present routing proposition requires a specific mechanism to deal with high load conditions, so a variation of the trunk reservation principles is proposed in Section 2.2. Simulation results obtained with RSR in the context of a fully meshed network are provided in Section 4. Section 5 gives an overview of implementation issues related to the computation of the routing tables used by RSR and concludes the paper.

2. Splitting the load

2.1. Computing path costs and path selection principles

We propose to use a composite path metric that reflects the load on the links and the path length for the estimation of the each path's quality. In this scope, load balancing consists in finding a tradeoff between resource conservation and load distribution. Actually, a network is likely to accept more connections if none of its parts are congested, but ensuring this should not lead to a global routing solution too far from optimal (when the network gets congested because much of the traffic is sent on "long" paths). In fact, optimality criteria for the routing solution are numerous, one of them being the usage of as little resources as possible (as in the multi-commodity flow problem), or maximizing a concave decreasing function of the edges loads as in [10]. In practice, RSR is strongly focused on preserving network resources while accepting as much load as possible. We consider the path metric:

$$M = \frac{r}{\exp(B \times h)}$$

where r is the available bandwidth on the path, and h its number of hops (links crossed). B is a parameter that sets the relative weight of bandwidth against length for the path's quality computation. This metric is derived from utility $U = \log(M)$ proposed in [12] specifically for the routing of elastic flows (see also[9]). In the remainder of this paper, we take $B = 1$, as it corresponds to a satisfactory compromise also chosen in [12].

2.1.1. Path selection

Once the path metrics are computed, the routing decision consists generally in choosing the best one, but it is also possible to split evenly the load on equivalent paths. This can be achieved by randomly selecting the paths with probabilities proportional to their estimated quality. Therefore, routing randomly according to M, paths with equal length and bandwidth receive the same amount of traffic, and the emphasis is strongly put on the shorter paths. With random routing, the load is readily split on all available paths, even before the routing protocol takes into account the changes in the link loads caused

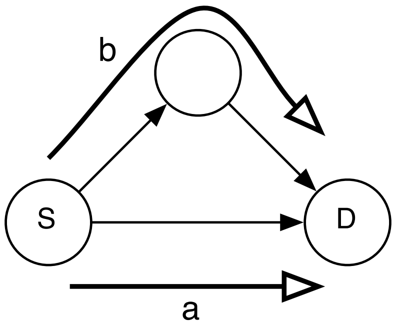


Figure 1. Route b is neglected when the routing policy selects the shortest path.

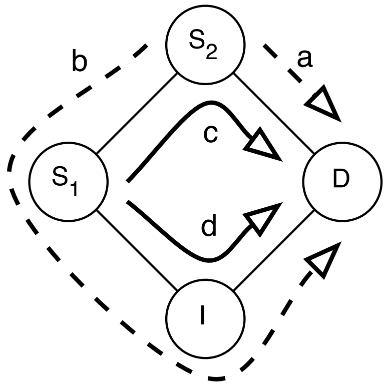


Figure 2. A problematic case for trunk reservation

by the presence of a new traffic. This is very valuable when a large number of connections enter the network at the same time, which occurs for instance when a component of the network goes down and the traffics that were using it are redistributed on the network. Variations of DAR (Dynamic Alternative Routing) [4] in fully meshed networks propose in the same manner to send traffic on alternate routes in proportions inversely proportional to the load that they already carry, but only if the direct one is saturated. We consider here that no route can be distinguished from the other ones, and in particular that no route can be identified as the "direct" one in contrast with the "alternate" ones for a given flow.

Random path selection in RSR follows a very simple principle. It incrementally builds the path, starting from the node of origin of the connection request. Then one of its neighbors is randomly selected and the same operation is reiterated from node to node. The probability of choosing a next hop n at the router a for a flow bound to d is :

$$P_n = \frac{M_n^{ad}}{\sum_{i \in N(a)} M_i^{ad}}$$

where $N(a)$ are the neighbors of a, and M_i^{ad} designates the metric associated with the *best path* from a to d starting with link $a - i$.

It can be noted that this incremental path construction is well suited for hop by hop dynamic routing. But it can also be used in the context of source routing, in which case the first node performs all the computation. This is the case for the simulations displayed in this paper. Such an approach requires that the adequate routing information be available at the source node, which is the case when a link state routing protocol is used.

2.1.2. Random routing on a subset of the paths

Random routing leads to a traffic distribution on the network very different from the one obtained with best path routing when using the same metric.

For instance, we consider the simple network shown in figure 1 composed of three similar links of bandwidth r, loaded by a set of connections from S to D. Using a single best path, route b does not receive any S-D traffic until the load on a reaches $l_a = r\ (1 - e^{-1})$. Then the routing starts to oscillate between the two paths. Using random routing, route b would receive its share of traffic from the beginning. In the case of a burst in the connection demands between S and D, this insures to balance the load before the routing algorithm takes into account the impact of the traffic on the link loads. But this behavior may be problematic in some cases, as it leads to the routing of a great proportion of the connections on sub-optimal paths. This can lead to an early congestion of the network, particularly when alternate paths are numerous like in fully-meshed networks [5]. Fortunately, it is straightforward to address this issue.

The path selection presented in 2.1.1 is slightly modified in order to have RSR to not use all feasible paths. No connection is routed on a path whose metric is under a given threshold (α times the metric of the best path, $0 < \alpha < 1$). The primary effect of this procedure is to limit the number of connections sent on saturated or long paths. For $\alpha = 1$, RSR degenerates to best path routing, and for $\alpha = 0$, plain randomized routing is used. If $N'(a)$ is the set of neighbors of a that are the starting points of paths complying to this condition:

$$P_n = \begin{cases} \frac{M_n^{ad} - \alpha \times M_{\max}^{ad}}{\sum_{i \in N'(a)} M_i^{ad} - \alpha \times M_{\max}^{ad}} & \text{if } n \in N'(a) \\ \\ 0 & \text{otherwise.} \end{cases}$$

In particular, for $\alpha > \exp(-3/2)$ (provided that $B = 1$) two routes sharing the same bottleneck both receive traffic only if their lengths differ by less than 2. Of course, if the shortest one becomes notably more loaded (if the bottleneck is not any more shared by the two routes) then the traffic is split. This setting also guarantees that the incremental routing of a connection be cycle-free, so it appears as a good lowest bound for this parameter (provided that the administrative metric of any link is greater than 1)[7].

2.2. Adapting trunk reservation to non fully meshed networks

Trunk reservation was proposed in the context of fully meshed networks, to address an intrinsic flaw of the dynamic routing algorithms in high load conditions. These algorithms were shown to oscillate between the global optimum which is that all nodes are using direct routes – and the local optimum when all consider that their direct routes are full and they seek overflow paths [8].

Trunk reservation consists in rejecting (overflow) calls that attempt to use a two-link route when the number of available circuits on one of its parts is below a given threshold. In this way, already highly loaded links keep accepting connections that follow a direct route, to the detriment of more resource consuming ones, and the global optimum is privileged. The order of magnitude of the threshold r_s is of 5% of the link bandwidth and the routing performance is rather insensitive to this parameter.

Applying trunk reservation in non-fully meshed networks in a straightforward manner is not satisfactory, mainly because the notions of direct and overflow calls can not be directly transposed to the general case. It appears that an initialization phase – or statical configuration – be required to indicate which routes should be protected and which ones should be privileged for each nodes pair.

For example, when considering the network appearing in figure 2, the application of trunk reservation is straightforward for the traffic between S_2 and D. In this case S_2

would not be allowed to send its traffic toward D on route b if the available bandwidth is below r_s. Between S_1 and D, there is no natural path, and the less demanded ones should actually be preferred. In fact, it is possible to use different trunk reservation parameters for different flows, *e.g.* giving in our example a greater priority on link $S_2 - D$ to the traffic between S_2 and D over the one between S_1 and D. However, finding the best (hierarchical) trunk reservation parameters is to our knowledge an open issue.

As it appears that trunk reservation should depend on the length of a route, and the existence of other more direct ones or not, the proposed mechanism is the following. A minimum r_{min} is assigned to the available bandwidth used and distributed by the routing algorithm (or "advertised by the links") when the actual number of free circuits is below this level.

In the context of this article, connections are definitively discarded and lost after two unsuccessful routing attempts.

3. Comparison with other routing strategies

3.1. Overview

The most commonly envisaged approach for routing protocols that can adapt to the links loading is most probably the Widest Shortest (WS) one. It is the routing strategy retained for QOSPF and recommended by PNNI. It is also proposed for traffic engineering with MPLS in [1]. With WS, a connection is sent on the shortest path that can handle it and, if more than one solution exists, on the path displaying the most available bandwidth. This approach stands as a good reference because it is actually used and does not require important changes in the routing protocols to be implemented.

Load balancing has also been a concern during the design of OSPF (version 2) [11], and ECMP (Equal Cost Multi-Paths) consists in using all the paths with equal administrative costs through a network. This mechanism does not rely on any load measurement.

Contrasting with WS on the complexity ground, MPLS-OMP [15] is aimed at the dynamic routing and load balancing of flows through a network. MPLS-OMP decreases the load on the most loaded links of the network, by gradually sending traffic on sub-optimal – alternate – paths (roughly determined on the basis of their administrative metrics). Simulations reported in [15] show the good behavior of this approach in realistic conditions, although the complexity of the load adjustment strategy makes it hard to implement. OMP also adapts well to varying traffic patterns, but its gradual moves from one equilibrium to another make it intrinsically less responsive than our proposal.

Cisco's EIGRP also uses of a composite path metric for load balancing. But EIGRP is based on a shortest paths computations, and it is in fact recommended to not use a dynamic metric, or at least one that only marginally depends on dynamic measurements. There is also no resource preservation associated to it.

3.2. Experimental results

In order to compare random shaped routing (RSR) to WS, our simulations focus on the source routing of fixed rate connections through a network whose links feature limited transport capacities. Source routing is used under the assumption that the exact and immediate state of each link is available at the source. This would not be true in more realistic conditions (because it takes time to broadcast the routing information in a

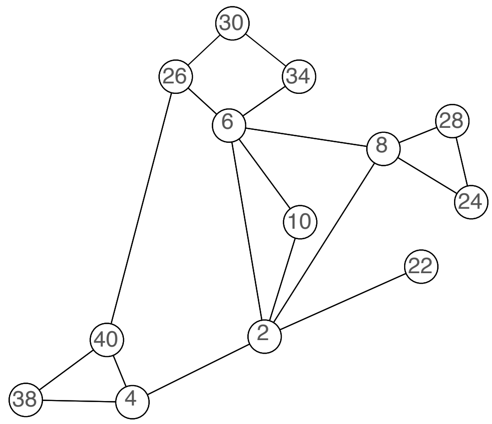


Figure 3. 13 nodes test network

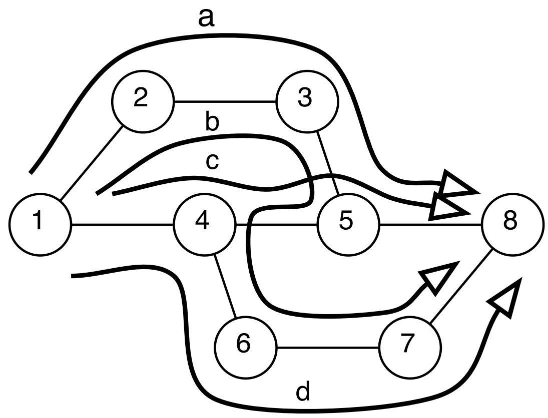


Figure 4. Routing solutions for demands between nodes 1 and 8

network), but this issue is out of the scope of the present paper.

In the case of two distinct paths that share at least a common first link, the measure used for the computation of the selection probability of this first link is the best one.

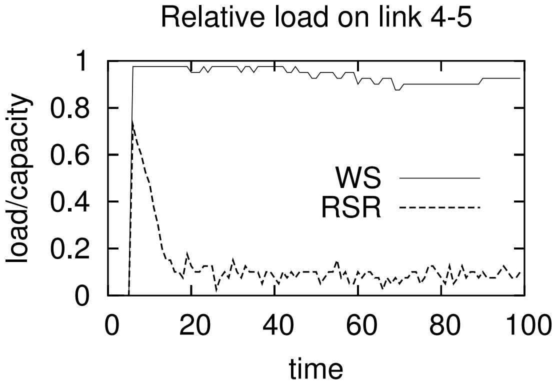
3.2.1. Competing flows

The context here is the one of a data network, loosely connected. Experiments take place in the 13 nodes network shown in figure 3, with links featuring a capacity of 20 simultaneous connections (except for the links on the path 2 - 10 - 6: 80, and 4 - 2: 40). The traffic is a Poisson process of intensity 50 triggering the creation of an average number of 5 connections lasting one time unit each with random source and destination nodes. The measures are averaged over 20 runs of 100 time units.

Table 1 shows the number of connections successfully routed through the network for various routing settings. The first observation is that without trunk reservation, the results are similar with RSR and WS. Also, it appears that not restricting the routing to the best path does not have any impact on the network's throughput. Secondly, using the resource preservation principles described in section 2.2 improves the network throughput significantly. The next paragraph details a particular case where RSR intrinsically outperforms WS.

Table 1
number of successfully routed connections for various routing algorithms

routing algorithm	av. num. of successful connections	std. deviation
WS	170.3	20.0
RSR $\alpha = 0.4$, $r_{min} = 0$	169.8	20.9
RSR $\alpha = 0.4$, $r_{min} = 3$	173.6	20.0
RSR $\alpha = 0.3$, $r_{min} = 1.2$	178.2	20.0



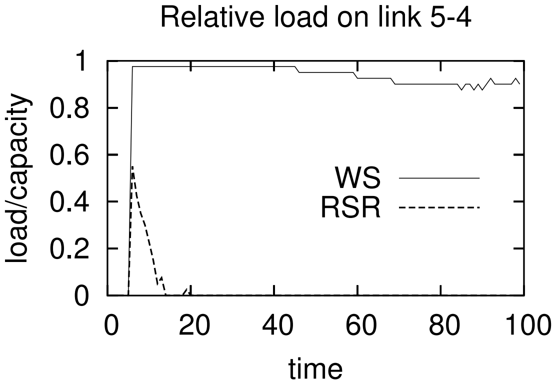


Figure 5. Load on links $4-5$ and $5-4$ for a high demand between nodes 1 and 8

3.2.2. Single flow

On the network shown in figure 4, we focus on the routing of a single commodity flow from node 1 to 8. The four available paths between 1 and 8 are depicted as a, b, c and d in figure 4. Under light load, and all links capacities being equal, the optimal routing is to use the direct route c. But if the traffic grows and c gets saturated, a part of the traffic must be sent on the two alternate routes a and d. Unfortunately, the WS strategy is not able to shift from the first load distribution to the second one, as it first saturates the direct route and then sends overflow traffic on route b through $1-2-3-5-4-6-7-8$. This strategy needs an adequate bootstrapping to reach the optimal routing, which can be the pre-existence of a traffic on link $4-5$ that causes it to be avoided from the beginning. On the contrary, RSR shifts gracefully from using only the direct route to using the two alternate routes as the load increases, up to a point where the direct route is totally neglected when the network capacity is reached.

Figure 5 illustrates these remarks: for a traffic from 1 to 8 that exceeds the capacity of the network (90 connections per time unit lasting 1 time unit for a link capacity of 40), the uni-directional links $4-5$ and $5-4$ (routes c and b) are fully loaded with the WS approach, but not used by RSR, which sends all the connections over the two secondary routes a and d. This lowers the overall network usage generated by the flows between 1 and 8, and in particular links $4-5$ and $5-4$ remain available to other traffics.

In this case, no trunk reservation of any type was used ($\alpha = 0.4$ and $r_{min} = 0$). It would cause in both cases a part of the traffic to be rejected instead of saturating alternate routes.

4. Comparison with DAR in a fully meshed network

The aim of the experiments carried out in this section is to test the RSR approach in a well-known context. Here RSR is not meant to lead to major improvements, as the usual approaches are satisfactory. This is rather a validation, having in mind the portability of RSR to other contexts.

DAR is a simple decentralized routing scheme, that was originally designed for the British Telecom trunk network [13]. It assumes a fully interconnected network, and uses only local information for taking routing decisions. When a call establishment request

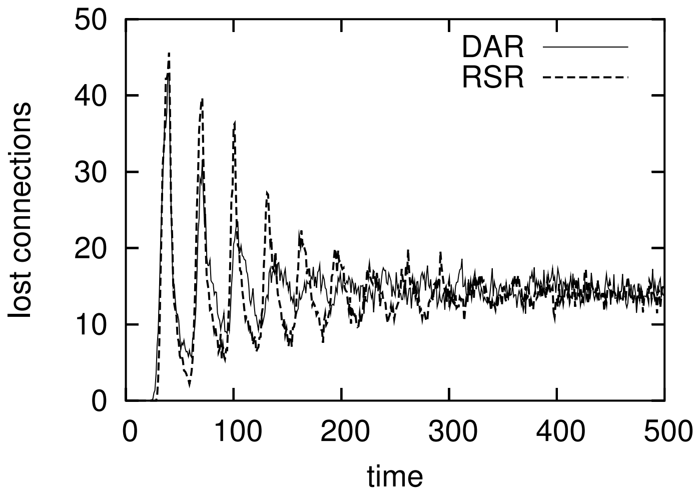


Figure 6. Number of lost connection establishment attempts per time unit for DAR and RSR

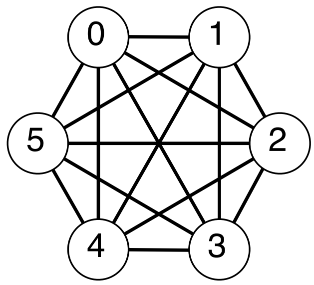


Figure 7. 6 nodes fully meshed network

reaches a switch, it always tries to select the direct route to the destination. If this link is saturated, the call request attempts to reach the destination *via* the default alternate switch known for this destination. If this two-link route is also busy, the call request is lost and a new default switch is chosen at random. Trunk reservation then consists in rejecting (overflow) calls that are attempting to use a two-link route when the number of available circuits is below a given threshold r_s.

4.1. Traffic discarding procedure

With the policy described in section 2.2, and in the particular case of fully meshed networks, connection dropping always occurs when a connection is sent on a path that cannot handle it. Three operating points are remarkable:

- connection dropping appears when the direct route is full, but it is nevertheless eligible because the available bandwidths r_{alt} on *all* alternate routes are low enough. *i.e.* $r_{alt} < r_{min} \times e^{-1}/\alpha$.
- all connections are lost when the direct route is full, and no alternate route are eligible. This means that $r_{alt} < r_{min} \times e^{-1} \times \alpha$ for all alternate paths.
- connection dropping also happens when an alternate route has no free circuit, but it is nevertheless used by the routing algorithm. This may occur when the bandwidth on the direct route r_{direct} is such that $r_{direct} < r_{min}/(e^1 \times \alpha)$. One can note that for $\alpha > e^{-1}$ this last case is avoided.

So it appears that this policy correspond to a behavior very similar to "normal" trunk reservation, with a gradually increasing connection dropping probability when the load on the links passes the point where a link is saturated and the other link loads are such that the first point above be valid.

Simulations confirm this remark, and RSR compares successfully to DAR with trunk reservation when applied in a 6 nodes fully meshed network (figure 7) under high load.

The number of lost connections is plotted in figure 6, for a "bandwidth" of 40 connections per link and a Poisson traffic of intensity 45 (each connection lasting 30 time units). The traffic gets concentrated in both cases to the direct routes, and RSR benefits from the *a priori* random distribution of the load on alternate routes, which explains its slightly better performance (on average, after stabilization, DAR looses 0.5 more connections per time unit). For DAR, a trunk reservation of 6 connections was used, $\alpha = 0.4$ and $r_{min} = 3$.

5. Conclusion

5.1. Implementation issues

RSR may be used for either hop-by-hop routing or source routing. Moreover, in the hop by hop case, RSR allows to greatly diminish the risk of loops while routing a connection as stated in [7].

In the simulations presented above, the routing computation is based on the true state of the network at the moment when a connection establishment request appears at its source node. However this computation is really based on past information as a delay is introduced by the routing protocols. In this respect, RSR presents two valuable features. First, *a priori* load balancing allows to distribute the load on the network before the routing protocols take into account the impact of this new traffic. Moreover, the load is distributed in a way that *tends* to maintain identical the traffic proportions associated to the different routes over time. Second, the two variables (available bandwidth and path length) required by RSR to compute its estimate of the route state can be gathered by most advanced routing algorithms.

The biggest concern with the load-adapting routing algorithms like the one presented here is their stability when the typical period of variations in the load are of the same order of magnitude as the algorithm convergence time. Although stability is an intrinsic feature of RSR, common damping techniques such as an hysteresis in the routing shifts should be considered. Agent-based routing [6] could also be considered as it was designed to keep track of more than one routing solution at a time in a dynamic environment.

5.2. Conclusion and future work

The approach presented in this paper is a simple and versatile routing policy that is designed to dynamically balance the load in communication networks. It does not require major implementation changes in the current routing protocols. Moreover, this approach is tunable and strongly focused on an efficient network usage through its adaptation of the trunk reservation principles. It gives results very similar to plain trunk reservation in fully meshed networks (Section 4), although it has been designed specifically for data networks, where it can significantly improve the network throughput and utilization efficiency (Section 3).

This work opens several new directions of future research. First, a stronger justification of the load routing proportions, which may lead to other metrics propositions can be studied. Second, the possible re-packing strategies that this approach can spawn are numerous. The most straightforward would be to discard and re-route connections residing on paths that are not any more considered as valid by the routing protocol. This should be done based on the acceptance parameters of the routing algorithm. In this way we will obtain a re-optimization method whose potential benefits are not known yet, but it responds to critical needs in the field of traffic engineering.

It would also be of great interest to investigate the application of RSR in other domains, like for example the routing of elastic flows, possibly with different performance estimates. Routing connections bulks is also a promising research domain, where the granularity of the flow splitting can be discussed with respect to the routing performance obtained.

REFERENCES

1. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for traffic engineering over MPLS. IETF RFC 2702. IETF, network working group, September 1999.
2. W. Ben-Amer, N. Michel, and B. Liau. Routing strategies for IP networks. 2000.
3. B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *INFOCOM (2)*, pages 519–528, 2000.
4. R.J. Gibbens. *Dynamic Routing in Circuit Switched Telecommunication Networks: The Dynamic Alternative Routing Strategy.* PhD thesis, University of Cambridge, 1988.
5. R.J. Gibbens and F.P. Kelly. Dynamic routing in fully connected networks. *IMA Journal of mathematical control & information*, 7:77–111, 1990.
6. M. Heusse and Y. Kermarrec. Adaptive routing and load balancing of ephemeral connections. In *IEEE ECUMN'2000*, pages 100–108, Colmar, France, september 2000.
7. M. Heusse and Y. Kermarrec. A new routing policy for load balancing in communication networks. In *ACS/IEEE International Conference on Computer Systems and Applications*, pages 267–272, Beirut, Lebanon, june 2001.
8. P.B. Key and G.A. Cope. Distributed dynamic routing schemes. *IEEE Communications Magazine*, pages 54–64, october 1990.
9. Q. Ma and P. Steenkiste. Routing traffic with quality of service guarantees in integrated service networks. In *NOSSDAV'98*, july 1998.
10. J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5), october 2000.
11. J. Moy. OSPF version 2. IETF RFC 2328, april 1998.
12. S. Oueslati-Boulahia and E. Oubagha. An approach to routing elastic flows. In *ITC16*, Edinburgh, 1999.
13. R.R. Stacey and D.J. Songhurst. Dynamic alternative routing in the British Telecom trunk network. In *ISS'87*, 1987.
14. G. Swallow. MPLS advantages for traffic engineering. *IEEE communications magazine*, 37(12):54 –57, december 1999.
15. C. Villamizar. MPLS-OMP. IETF draft, 1998. work in progress.