The MMF rerouting computation problem

A. Bashllari*, D. Nace*, E. Gourdin, O. Klopfenstein****

**Laboratoire Heudiasyc UMR CNRS 6599, Université de Technologie de Compiègne*

60205 Compiègne Cedex, France, {abashlla, dnace}@hds.utc.fr

***France Telecom $R\&D$, 38-40 rue du Général-Leclerc,*

92794 Issy-les-Moulineaux, France {eric.gourdin,olivier.klopfenstein}@orange-ftgroup.com

Abstract

In this paper, we present the max-min fair (MMF) rerouting computation problem for different end-to-end rerouting strategies with respect to simple link failures. This work is in continuation of our earlier works on MMF [4, 5]. We present a general iterative algorithm for the MMF rerouting problem, as well as the associated mathematical formulations for different rerouting strategies. We focus on the end-to-end rerouting with stub release and shortly discuss some related complexity results.

Keywords: Max-Min Fairness, rerouting, multi-commodity flow, linear programming, path generation.

1. Introduction

We focus in this paper on the max-min fair (MMF) rerouting computation problem. This problem corresponds to computing a feasible routing such that the minimal demand satisfaction ratios vector associated with the set of simple link failures is MMF, which is termed below "MMF-RR vector". We consider the case when only links are subject to (non-simultaneous) failures. Each component of the vector is associated with the minimum demand satisfaction ratio over all traffic demands achieved after the restoration process, for a link failure state[1] and a given end-to-end rerouting strategy. Before describing in detail how this problem can be solved, we briefly put the paper in its context. This study is thus concerned with two aspects, first the network resource allocation and second, the max-min fairness. With respect to network resource allocation problems, we will cite only a few works closely related to our problem. First, in [3], the authors propose models for dimensioning a survivable network with end-to-end rerouting with stub release, that is *network resources used for traffic routing, that are released after a failure, could also be used for traffic rerouting*. They give the proof of the NP-hardness of the associated pricing problem. This problem is also studied in [7], where another NP-hardness proof is given. Worth mentioning is also [6], where the authors are the first to put the survivable network design problem (under local rerouting) in a MMF perspective. In contrast, we consider in this paper the satisfaction ratio of traffic demands in link failure states and deal with end-to-end rerouting strategies, known to ensure a more efficient resource utilization although reputed to be computationally harder. On the other hand, the max-min fairness and lexicographic optimization with applications to networks have also been object of a lot of works these last decades; see for instance [6, 8, 9] and the references therein. Finally, this work is in continuation of our earlier works on MMF [4, 5].

The paper is organized as follows. In Section 2, after some preliminaries, we show how the MMF rerouting computation problem can be solved. In Section 3, we focus on end-to-end rerouting with stub release strategy and discuss some complexity issues. We also provide a brief discussion for two other rerouting strategies, namely end-to-end rerouting without stub release and global rerouting. Some numerical results are reported in Section 4 and finally in Section 5, we conclude.

2. The MMF rerouting computation problem

Before describing how the MMF-RR vector can be computed, we will recall some mathematical notions, definitions and notation useful for the remainder of the paper.

Preliminaries

Let us recall first the definition of MMF.

Definition 1. *A vector $x \in X \subset R^n$ is max-min fair on X if, and only if,* $\forall y \in X,\ \exists\, a \in \{1, ..., n\},\ (y_a > x_a) \Rightarrow (\exists\, j \in \{1, ..., n\},\ y_j < x_j \le x_a)$.

[1] Failure state is defined as a network state where some hardware, in our case a link, has ceased to function. Conversely, nominal state is defined as the state where all network hardware is operational (without failure).

Recall that MMF is closely related to lexicographic optimization [9]. Generally speaking, lexicographic optimization arises when one needs to optimize beyond a simple criterion objective function, for which several or an infinity of solutions exist. For example, consider a set of end-to-end traffic demands. The first objective could be to maximize the minimum demand throughput among all demands. Since an infinity of solutions may exist, it will be interesting to find a solution which also maximizes the second smallest demand throughput. Thus, the optimization can be continued until we find a solution with the following property: there is no other solution which gives a better value for a traffic demand without decreasing the value of a traffic demand less served. This is known as max-min fairness property. The relation between both notions, that is MMF and lexicographic optimization, is studied in [9], where it is shown that in any compact convex set X, there always exists a MMF vector x such that its corresponding sorted non-decreasing values vector is lexicographically maximal in X.

With respect to traffic rerouting, various strategies exist to withstand single link failure. We consider here three rerouting strategies: the partial end-to-end rerouting with/without stub release, and global rerouting. For the first two strategies, only the interrupted traffic may be rerouted (i.e., *partial*), in contrast to global rerouting which allows the rerouting of all traffic. The partial end-to-end rerouting with stub release corresponds to the case where network resources used for traffic routing, can be released after a failure, and reused for traffic rerouting. In the same spirit, we call "without stub release" the end-to-end rerouting strategy when network resources are used exclusively either for the nominal routing or for the links failure rerouting.

Let us make now some hypotheses on routing. As the network manager needs to ensure a "quality" routing, satisfying a minimum traffic value for all demands in the nominal state could be imposed. Assuming that the network resources allow a feasible routing, such a ratio could be the maximal value that can be achieved. After some notation, we give below a mathematical model, called the nominal state problem (NS), intended to compute such a ratio. Next, we give the general approach for the MMF rerouting computation problem.

Notation:

- A telecommunication network is represented by an undirected graph $G = (V, A)$. It is made up of a set of $|V| = n$ routers and $|A| = m$ links, with capacities denoted by C_a, (link a).
- Let D be the set of traffic demands d. Each demand is defined by a pair of extremity nodes and a non-negative traffic value T_d.
- Let $h^0(d)$ $(h^l(d))$ be the set of paths of graph G (respectively $G^l = (V, A \setminus \{l\})$) between the extremities of the demand d.
- Let $x_{j,d}$ (respectively $x^l_{j,d}$) be the traffic part routed on path j for demand d in nominal state (respectively failure state l). We use the notation $a \in j$ to indicate that link a belongs to the routing path j.
- Let λ_0 be the minimum demand satisfaction ratio in nominal state.
- Let $\lambda = \{\lambda_1, \lambda_2, ..., \lambda_{|A|}\}$ be the solution vector where λ_l is the minimum demand satisfaction ratio in failure state $l \in A$. Let $\overrightarrow{\lambda}$ denote the corresponding vector sorted through non-decreasing values.

With respect to the notation given above, $x_{j,d}$, $x^l_{j,d}$ and vector λ correspond to variables in our mathematical formulation. The (NS) problem can be modeled using an arc-path LP formulation as follows:

$$\max \ \lambda_0$$

$$(\pi_a) \quad \sum_{j \in h^0(d), d \in D, a \in j} x_{j,d} \leq C_a \quad \forall a \in A \tag{1}$$

$$(\pi_d) \quad -\sum_{j \in h^0(d)} x_{j,d} \leq -\lambda_0 T_d \quad \forall d \in D \tag{2}$$

$$x_{j,d} \geq 0 \quad \forall d \in D, \forall j \in h^0(d) \tag{3}$$

where constraints (1) are capacity constraints, and constraints (2) are demand traffic satisfaction constraints. To solve the (NS) problem we use the column generation method, which, in our case, corresponds to the shortest path computation on the initial graph valued by dual coefficients.

An iterative approach for computing the MMF-RR vector

Let us present the problem more precisely: we associate with each link failure the minimum rerouting satisfaction ratio among all traffic demands, that is for each failure state l, we consider the satisfaction ratio obtained after restoration for all traffic demands $d \in D$ and note it with λ_l. This will be represented as a single component in the MMF-RR vector. Now, the problem can be stated: *Given a network, a capacity vector and a set of traffic demands, compute a feasible routing such that the minimum rerouting satisfaction ratios vector is MMF.*

The above problem is solved through an iterative algorithm, which allows computing at least one component of the vector per iteration. Hence, an iteration is basically concerned with solving a linear programming problem, called below $P_R(L)$, which depends on the chosen rerouting strategy. More precisely, the first iteration is intended to compute the largest minimum satisfaction ratio λ' with respect to all considered failure states and identify the failure state l, where this ratio is obtained. Next, we set for failure l the minimum satisfaction ratio to λ' and compute the second largest minimum satisfaction ratio (also noted λ') over the remained link failure states. The algorithm is summarized below:

Algorithm MMF-RR

1. *Compute the value of λ_0 (solve the (NS) problem)*[2].
2. $Set\ L = \emptyset$.
3. $While\ L \neq A\ do$: *solve $(P_R(L))$ with respect to the restoration strategy* (compute λ').
4. Fix $\lambda_l = \lambda'$, for all l such that the minimum demand ratio cannot be further increased (according to the MMF principle); set $L = L \cup l$. *Return to 3.*

Above, L is the set of index link failures with λ_l already computed.

3. End-to-end rerouting with stub release strategy

As mentioned above, the problem $P_R(L)$ is formulated with respect to the considered restoration strategy. We will focus below on the case of end-to-end rerouting with stub release (WR). We obtain the following problem $(P_{WR}(L))$:

$$\max \quad \lambda'$$

$$(\pi_a^0) \qquad \sum_{j \in h^0(d), d \in D, a \in j} x_{j,d} \leq C_a \qquad \forall a \in A \tag{4}$$

$$(\mu_d^0) \qquad - \sum_{j \in h^0(d)} x_{j,d} \leq -\lambda_0 T_d \qquad \forall d \in D \tag{5}$$

$$(\pi_a^l) \qquad \sum_{j \in h^0(d), d \in D, l \notin j, a \in j} x_{j,d} + \sum_{j \in h^l(d), d \in D, a \in j} x_{j,d}^l \leq C_a \qquad \forall a, l \in A \tag{6}$$

$$(\mu_d^l) \qquad - \sum_{j \in h^0(d), l \notin j} x_{j,d} - \sum_{j \in h^l(d)} x_{j,d}^l \leq -\lambda_l T_d \qquad \forall d, l \in L \tag{7}$$

$$(\mu_d^l) \qquad - \sum_{j \in h^0(d), l \notin j} x_{j,d} - \sum_{j \in h^l(d)} x_{j,d}^l \leq -\lambda' T_d \qquad \forall d, l \in A \setminus L \tag{8}$$

$$x_{j,d} \geq 0, x_{j,d}^l \geq 0 \qquad \forall d \in D, \forall l \in A, \forall j \in h^l(d). \tag{9}$$

Constraints (4) and (5) are nominal routing constraints and the others are link failure rerouting constraints, respectively capacity constraints (6) and traffic constraints (7, 8). The term in the left hand side gives the associated dual coefficient. With respect to the dual formulation of $P_{WR}(L)$ it can be remarked that there is at least one $l \in A \setminus L$ and one $d \in D$ such that $\mu_d^l > 0$. This allows identifying the link l to be introduced in L while fixing $\lambda_l = \lambda'$.

Complexity issues: The above problem can be solved using the column generation method where the new columns represent candidate paths. We can remark that the reduced cost associated with each column (path $j \in h^0(d)$) is given by $\sum_{a \in j} (\pi_a^0 + \sum_{l \notin j} \pi_a^l) - \sum_{l \notin j} \mu_d^l - \mu_d^0$ (!). The pricing problem associated with $(P_{WR}(L))$ has been studied by Maurras and Vanier in [3] and Orlowski in [7]. They consider a general formulation of the above pricing problem[3], and show its NP-completeness. More precisely, they show the equivalence of a general formulation of the pricing problem with respectively the decision Hamiltonian chain problem in [3] and the max-cut problem in [7]. However, it can be shown that the pricing problem (consequently the $P_{WR}(L)$ problem) can be solved in polynomial time for the case of a fixed l link failure.

[2]This step is optional since λ_0 could be imposed by the network manager.

[3]Indeed, no considerations on the relation among dual coefficients involved in (!) are taken into account and therefore no conclusions can be drawn on the complexity of the initial problem $P_{WR}(L)$ itself.

The associated pricing problem can be reduced to computing the shortest path respectively in graph $G \setminus l$ and G passing through l:

Proposition 1. *The pricing problem of* $(P_{WR}(L))$ *can be solved in polynomial time for the case of a fixed single link failure.*

Proof. Let denote with l the only link failure to be taken into account. Then, the general shortest path problem $\sum_{a\in j}(\pi_a^0 + \sum_{l\notin j}\pi_a^l) - \sum_{l\notin j}\mu_d^l$ can be solved easily by considering two cases, depending on $l \in j$ or not. In the case when $l \notin j$, we compute the shortest path in $G \setminus l$ for weights $\pi_a^0 + \pi_a^l$ on links $a \in A$. In the other case, i.e. when $l \in j$, we need to compute the shortest path passing through l in G valued with π_a^0. Clearly the first problem is a polynomial one while the second can be reduced to the problem of computing (in undirected graphs) a pair of vertex-disjoint paths of total minimum link cost, already solved by Suurballe in [10]. Indeed, it suffices to add two dummy nodes, the first one (called s) connects the extremity nodes of the demand d and the second node (called t) connects the link extremities of l, and compute a pair of vertex-disjoint paths of total minimum link cost between s and t. Then, clearly each shortest path obtained by the Suurballe's Algorithm, relies separately one extremity of the failed link l with one of extremities of the demand d. Joining these two paths and link l allows obtaining the shortest elementary path passing through link l. ♦

Remark. The above reasoning does not hold for the directed network case. Indeed, computing a pair of vertex-disjoint directed paths of total minimum link cost is already proved to be NP-complete, [1].

Two other rerouting strategies

In this section we focus on two other end-to-end rerouting strategies, which are the (partial) end-to-end rerouting without stub release and global rerouting.

End-to-end rerouting without stub release (WoR): The mathematical formulation of the associated rerouting problem, called $(P_{WoR}(L))$, is obtained by replacing in the $(P_{WR}(L))$ formulation the constraints (6) with (10):

$$\sum_{j\in h^0(d),d\in D,a\in j} x_{j,d} + \sum_{j\in h^l(d),d\in D,a\in j} x_{j,d}^l \le C_a \ \ \forall a,l \in A \tag{10}$$

If we solve the $(P_{WoR}(L))$ problem using the column generation method, we can remark that the reduced cost associated with each path $j \in h^0(d)$ is given by $\sum_{a\in j}(\pi_a^0 + \pi_d^a + \sum_{l\in A}\pi_a^l) - \sum_{l\in A}\mu_d^l - \pi_d^0$. Clearly such paths can be computed in polynomial time.

Global rerouting (GR): In the case of global rerouting (GR), the problem becomes simpler. We remark that each link failure situation leads to a feasible routing problem maximizing the minimum traffic satisfaction ratio, which is exactly defined as the (NS) problem applied to network $G \setminus l$. Thus, there is no need for applying the iterative algorithm presented above. Each component is computed through a separate program.

4. Computational results

The computational results presented in this section concern the calculation of traffic satisfaction ratios vector for the three end-to-end rerouting strategies studied earlier (global, with and without stub release). The approaches were implemented in C++ using CPLEX 8.0. All tests were run on a machine with the following configuration: Windows XP, 1 processors Pentium 4 1.4GHz, 512 MB of RAM. In Table 1 we summarize the main characteristics of four network instances.

Network	NET_6	NET_11	NET_15	NET_26
Nodes	6	11	15	26
Arcs	11	25	30	43
Demands	15	55	105	264

Table 1: Networks description.

We show in Figure 1 (respectively Figure 2) the results obtained for the NET_26 network (respectively NET_11). For the two other networks the results are similar. As expected, the global rerouting strategy performs better than other ones. On the other hand we can notice that end-to-end rerouting with stub release gives slightly better results than end-to-end rerouting without stub release for the first components of the rerouting ratios vector. But the curves cross each other, and end-to-end rerouting without stub release provides better rerouting ratios for last components. This illustrates the fact that both strategies have at a certain extent similar performances.

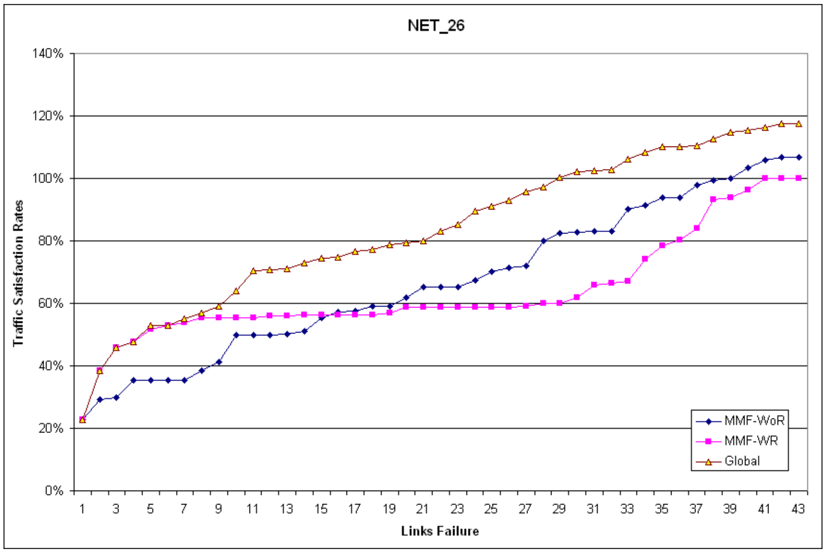

Figure 1: Traffic satisfaction ratios vector NET_26.

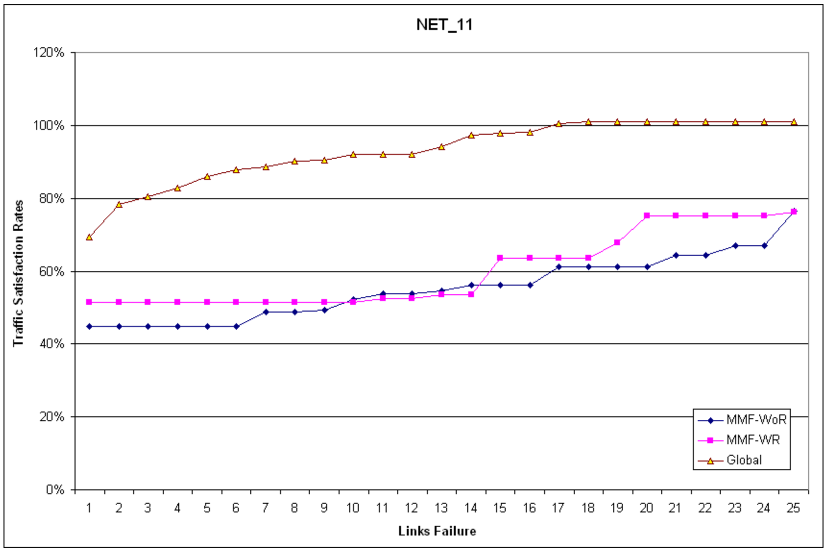

Figure 2: Traffic satisfaction ratios vector NET_11.

In Table 2 we present the calculation time of $\overrightarrow{\lambda}$ (MMF rerouting ratios vector) for the three end-to-end rerouting strategies studied above, implemented in NET_6, NET_11, NET_15 and NET_26 networks.

Network	NET_6	NET_11	NET_15	NET_26
T_MMF-WR	0.21	73.846	1041.85	6700.61
T_Global	0.25	0.631	1.272	5.548
T_MMF-WoR	0.19	9.023	11.166	869.08

Table 2: Calculation time (in sec.) of $\overrightarrow{\lambda}$.

5. Conclusion

In this paper, we have presented an iterative algorithm for computing the MMF rerouting ratios vector for different end-to-end rerouting strategies with respect to link failures. We also provide some discussion on complexity of the end-to-end rerouting with stub-release case. The proposed approach has been tested in medium-size networks going till 26 nodes and more than 40 links and the overall computing time varies from a few seconds to hours.

References

[1] S. Fortune, J.E. Hopcroft and J. Wyllie, The directed subgraph homeomorphisme problem, *J. Theoret. Comput. Sci. 10 (1980), 111-121*.

[2] M. R. Garey, D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*, (W. H. Freeman 1979).

[3] J.-F. Maurras and S. Vanier, Network Synthesis under Survivability Constraints, *4OR, Quarterly Journal of Belgian, French, and Italian Operations Research Societies*, 2(1), 2004, 53-67.

[4] D. Nace, O. Klopfenstein,“On the lexicographically minimum loaded networks”, INOC, 2005, p. 776-782

[5] D.Nace, L.N.Doan, E.Gourdin, B.Liau, “Computing optimal max-min fair resource allocation for TCP flows”, *IEEE Transaction on Networking, Vol. 14, No. 6, p. 1272-1281, Dec 2006.*

[6] P. Nilsson, M. Pioro, Z. Dziong, Link Protection within an Existing Backbone Network *Proc. of INOC 2003*, Evry, 2003, 435-441.

[7] S. Orlowski, *Local and global restoration of node and link failures in telecommunication networks*, Diploma thesis, Konrad-Zuse-Zentrum fr Informationstechnik Berlin (ZIB), 2003, Berlin, Germany.

[8] M. Pioro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*, (Morgan Kaufmann Publishers, 2004).

[9] B. Radunovic, J.-Y. Le Boudec, A Unified Framework for Max-Min and Min-Max Fairness with Applications, *Proc. of 40th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, October 2002.

[10] J.W. Suurballe, Disjoint paths in a network, *Networks 4 (1974), p. 125-145.*