On-line Routing and Bandwidth Allocation for Elastic Traffic and for its Restoration

Péter Laborczi[a] Tibor Cinkler[b]

[a]Bay Zoltán Foundation for Applied Research, H-1116 Budapest, Fehérvári út 130.,
E-mail: laborczi@ikti.hu, Phone: +36-1-463 0510 Fax: +36-1-463 0506

[b]High Speed Networks Laboratory, Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics, Magyar tudósok krt. 2, H-1117 Budapest, Hungary
E-mail: cinkler@tmit.bme.hu, Phone: +36-1-463 1861 Fax: +36-1-463 1763

***Abstract*— In infocommunications networks the amount of free network resources varies in time significantly. Furthermore, after a network failure a large amount of backup paths are built up and later torn down. To achieve the highest possible network throughput and availability while keeping the loss of data low and guaranteeing fairness, the bandwidth (rate) of sources should be adjusted accordingly in an elastic way. After a complex network failure the path of the demand should be changed (re-routed) as well as its bandwidth likely decreased.**

Existing papers study the case when the traffic matrix is given in advance and centralised management is assumed. In contrast to these we investigate the case of on-line (i.e., dynamic) routing when unpredictable backup or new demands arrive in an unforeseen sequence. For this purpose instead of the centralised management we use distributed control. We propose flexible methods that determine the bandwidth of elastic sources *and* **the routes used by these demands** *simultaneously***. We evaluate the algorithms according to the obtained blocking ratios, network utilisation and computational time. We show that they yield lower blocking ratio and higher network utilisation than conventional shortest path methods.**

I. INTRODUCTION

Nowadays, in infocommunications networks the data-rate of sources and the arrival rate of new demands typically varies in time significantly. This causes a significant variation of link loads. After a link failure the restoration of connections can be realized within this elastic bandwidth. To guarantee higher throughput (better resource utilisation) and higher availability it is advantageous to make these sources able to adapt their rates to actual network conditions. Both in high priority real-time and in non-real-time traffic of the Internet there is a growing interest in devising bandwidth sharing algorithms [1] which can cope with a high bandwidth utilisation and at the same time maintain some notion of *fairness*, such as the Max-Min (MMF) [2], [3] or Proportional Rate (PRF) [4] fairness.

Three variants can be applied for determining elastic protection paths: (1) *fixed paths*, (2) *pre-defined paths*, or (3) *free paths*. In the case of *fixed paths* there is a single path defined between each origin-destination (O-D) pair and the allocation task is to determine the bandwidth assigned to each demand. In the case of *pre-defined paths* between each O-D pair, there is a set of admissible paths that can be potentially used to realize the flow of the appropriate demand. In this case the allocation task does not only imply the determination of the bandwidth of the flow, but also the identification of the specific path that is used to realize the demands [5]. In the case of *free paths* there is no limitation on the paths, i.e., the task is to determine the bandwidth of the traffic AND the routes used by these demands simultaneously. This novel approach, the joint path and bandwidth allocation for elastic traffic is the main topic of this article.

Recent research results indicate that it is meaningful to associate a minimum and maximum bandwidth with elastic traffic [6]. For a backup path the lower bound (minimum bandwidth) should be set to the guaranteed bandwidth of the demand while the upper bound (maximum bandwidth) to the actual size of it. Consequently, it is important to develop models and algorithms for such future types of networks. For the bounded elastic environment we propose a special weighted case of the MMF notion, called *Relative Fairness* (RF) that maximises the minimum rates relative to the difference between upper and lower bounds for each demand.

Considering literature different aspects of the max-min fairness policy have been discussed in a number of papers, mostly in ATM ABR context, since the ATM Forum adopted the max-min fairness criterion to allocate network bandwidth for ABR connections, see e.g. [7], [8]. However, these papers do not consider the issue of path optimisation in the bounded elastic environment. MMF routing is the topic of the paper [9], where the widest-shortest, shortest-widest and the shortest-dist algorithms are studied. These algorithms do not optimise the path allocation at all. A number of fairness notions are discussed and associated optimisation tasks are presented in [6] for the case of unbounded flows and assuming fixed routes.

Proportional Rate Fairness (PRF) is proposed by Kelly [4] and also summarised by Massouline and Roberts in [6]. The objective of PRF is to maximise the sum of logarithms of traffic bandwidths that increases the throughput while it dete-

---

The work of Peter Laborczi has been partially supported by a NAP project of the National Office for Research and Technology, by the Bolyai János Researchers Fellowship of the Hungarian Academy of Sciences and by the OTKA grant T42559.

The work reported in this paper has been supported by the European 6FP IST project IP NOBEL II (http://www.ist-nobel.org).

riorates the fairness by prioritising the shorter paths. Recent research activities focused on allocating the bandwidth of fixed paths. In [5] the approach has been extended in such a way that not only the bandwidth but also the paths are chosen from a set of pre-defined paths. In contrast instead of pre-defined path sets [10] proposes methods with "free paths", i.e., any path can use any available link. In [12] a new approach is described for asynchronous distributed rate control for elastic sessions. The MMF is generalised as "MMF Sharing with Minimum Session Rates". However, in this paper all sessions and their routes are given and static. In [11] algorithms for various fairness definitions are proposed.

Here we propose a new algorithm for the on-line routing and bandwidth allocation for elastic traffic for both, working an backup paths. Our objective is twofold. First, to develop an algorithm that is able to solve the dynamic connection establishment problem in an optimal way, where the network performance is measured by the blocking probability and network utilisation. Second, to give a detailed analysis of the allocation strategies applying different network configurations.

We assume that each node of the network is aware of the following: the topology of the network (of the considered domain); for each link the link capacity, link usage level, link control parameter (LCP_l meaning the highest bandwidth usage per demand on link l); and furthermore the node is aware of all the paths and sizes of sessions that originate from it. QOSPF (Quality of Service Open Shortest Path First) can be extended by adding three more Link State Update units to advertise the above link parameters. RSVP is also capable of supporting these requirements. The method can be applied for the Available Bit Rate (ABR) Service Class in Asynchronous Transfer Mode (ATM) Networks in a similar way.

Every switch can modify the link control parameters (LCP) of connected links causing automatically changing the session rates. In QOSPF an attempt to trigger an update of these information is done when the parameters differ significantly from the previously advertised value.

We assume that the sessions (working or backup paths) are built up and torn down dynamically according to users requests. Two cases are studied: in the first one the paths of the demands are static during their living period, just their bandwidth can be tuned. This is called *Semi-Elastic Traffic* (or *Re-Routable Elastic Traffic*) in this paper. In the second one, the path of the demand can be arbitrarily changed later to ensure highest throughput. This is called *Full-Elastic Traffic* in this paper.

II. ALGORITHM FOR SEMI-ELASTIC TRAFFIC

In this section the proposed algorithm is described for Semi-Elastic traffic, i.e. the case when to active sessions may not be re-routed. In this case we determine both the path and the bandwidth of the incoming (working or backup) demands. In each time interval either one new demand request emerges or an existing connection is torn down according to the Discrete Event Simulation (DES) principles. In the first case the request is either accepted, or rejected (blocked). The demand is accepted with a route and a certain bandwidth associated with it if there is a path such that on each link of it the sum of the lower bounds does not exceed the capacity of the link. Otherwise, it is blocked.

A. Routing

In this subsection we study the route selection method of one incoming demand. We use Dijkstra's shortest path algorithm and a shortest-widest path algorithm, optimising a linear combination of the number of links along a path *and* the width of the path. The width of a path is the bandwidth that can be allocated along the path. For the shortest path algorithm five weight functions have been employed and compared, which have best performed in the study of [14]. Let w_l denote the weight, C_l the capacity of link l and λ_l is the allocated bandwidth on it. All saturated links ($C_l = \lambda_l$) are temporarily deleted to avoid division by zero.

1. $w_l = 1$ for all links. This is the simple fixed, shortest path routing, which is used as the reference to compare more advanced weight functions to.
2. $w_l = \lambda_l$ Using this function the demand is routed onto those links, on which the used bandwidth is the smallest.
3. $w_l = \frac{1}{C_l\left(C_l - \lambda_l\right)}$ In case of this function the weight of a heavy loaded link will be very large (the denominator of the formula is close to zero), therefore these links will not be loaded further if there is at least one less loaded path for the demand.
4. $w_l = \frac{C_l}{C_l - \lambda_l}$ This function is very similar type to the previous one, but tests show that the difference between them is quite significant.
5. $w_l = \frac{1}{C_l - \lambda_l}$ This is a little bit simplified function, but in spite of this it seems to be very effective.

Besides Dijkstra's algorithm, a more advanced routing strategy has been worked out, that optimises the linear combination of the path length and of the width of the path. This is based on Integer Linear Program (ILP) minimising the length of the path while maximising the width of it.

The route determination can be done in two approaches: in a $pessimistic$ or in an $optimistic$ way. First, assume that the objective is to minimise the blocking probability. In this case the weights are set according to the free space when each session gets its minimum bandwidth (i.e., λ_l is the sum of minimum bandwidth of demands using link l). This is called pessimistic strategy since it assumes that some links will be overloaded. In this case we calculate with minimum sizes of the demands. Second, assume that the objective is to maximise the sum of link control parameters (LCP_l). Then the weight is set according to the amount of actual free capacity (λ_l is the sum of actual bandwidth of demands using link l). This is called optimistic strategy since it does not consider that some links can be overloaded. In this case we calculate with actual sizes of the demands.

B. Bandwidth Allocation: Iterative Capacity Setting

Besides the routing strategy, the other task is to allocate bandwidth in a fair way when a route has been determined for the new demand request. After a route for a new demand has been built up, or an existing one torn down, the link control parameters along their paths are adjusted by the following method causing that the rate of some sources will be decreased ("compressed"). The fact that some demands are compressed, causes that capacity on some other links might be freed. Consequently, some other demands can be expanded accordingly. Expanding demands iteratively raises running time, but improves the quality of the solution, in sense of total allocated bandwidth and network utilisation, but not in sense of blocking ratios, which remains unchanged. This method is called Iterative Capacity Setting.

The basic principle of this algorithm is also outlined in [12] for the MMF case. Here, we extend it for the RF fairness definition as well. The link control parameters are sent to the source of the demand in the corresponding RM cells of an ABR service over ATM, or are advertised by the QOSPF protocol in an IP network.

Each path may have one or more bottlenecks. We informally define the set of bottlenecks B_p of a path p as the minimum set of links B_p with the property that if we would increase the capacity on each link $l \in B_p$ then the bandwidth of the session could be increased. It is obvious that the actual capacity of a session is determined (maximised) by the set of its bottlenecks, i.e., we calculate its rate according to the bottlenecks. The method consists of three phases:

- *Initialisation Phase.* For every link l its source node calculates the link control parameters LCP_l^d for every demand d on a local basis as follows: $LCP_l^d = C_l/\delta_l$, where δ_l denotes the number of demands on link l. During the resource reservation procedure a control message is sent along the path from the source to the destination. Each node analyses the actual allowed bandwidth of demand d (b_d) carried in the control message. This value is initially set by the source node of the demand to the numerical equivalent of infinity. Whenever LCP_l^d is smaller than the value of the bandwidth reservation field b_d of the control message, then b_d will be set to LCP_l^d, i.e., $b_d = LCP_l^d$. When this message arrives to the destination of d, the maximum allowed bandwidth will be the lowest value of LCP_l^ds along the considered path. Next the allocation message from destination to the source will allocate the resources hop-by-hop.
  The above described method leads to MMF fair rates. However, some links can be underutilised because of the following reason. Consider a connection that has two bottleneck links l_1 and l_2. If $LCP_{l_1}^d < LCP_{l_2}^d$ then the bandwidth of this link will be $LCP_{l_1}^d$, while on l_2 some unused capacity will remain. This can be partitioned among other $\delta_l - 1$ connections, since at least one of the affected connections will not need more capacity on this link.
- *Increment Bandwidth Phase.* If a link controller realizes that less capacity is reserved than enabled (this is because of a bottleneck at an other link), this capacity can be used for other demands. Given a link l whose capacity is not fully utilised ($b_d \leq LCP_l^d$, $\forall d, l$), we increase the link control parameter for that link until the capacity is fully utilised.
  $$LCP_l^d += (C_l - \lambda_l)/(\delta_l - 1)$$
  where λ_l denotes the total reserved capacity on link l, $\lambda_l = \sum_d x_l^d b_d$. If it is known that there are more than one, e.g., δ_l' demands having bandwidths limited by other bottlenecks and not by link l, we will use $(\delta_l - \delta_l')$ as the denominator.
- *Decrement Bandwidth Phase.* If the amount of free network resources decreases then the total flow through a link may exceed its available capacity. This case is handled by decreasing the link control parameter to maintain feasibility as follows:
  $$LCP_l^d = C_l/\delta_l$$

This bandwidth allocation yields bandwidths that are MMF fair with unbounded demands. Let m_d and M_d denote the lower and upper bound of demand d. For $m_d \neq 0$, and for the RF fairness definition, the algorithm works as follows. In the initialisations phase each session gets its minimum bandwidth, and the remaining link capacity in a weighted manner:

$$LCP_l^d = (C_l - \sum_d m_d)/(M_d - m_d),$$

according to the principle of RF fairness definition. This bandwidth allocation yields bandwidths that are RF fair.

III. ALGORITHM FOR FULL-ELASTIC TRAFFIC

In this case the paths can be arbitrarily modified (re-routed) even while the sessions over them are alive (active). The advantage of this approach is that it is allowed to rearrange (re-route) the paths that originate at the same node followed by the new demand allocation. This decreases the blocking and/or increases the throughput. However, it can be applied only for connection that allow re-routing, since re-routing can lead to delay variations and packet re-ordering. As an example IP traffic allows re-routing, therefore an IP over MPLS or IP over ATM allows such re-routing.

The algorithm first builds a graph model in which it temporally removes all demands originating from the source of the new demand s and then runs a minimal cost single-source multicommodity flow subroutine [13] for the previously removed demands and the new demand s. Finally, it applies the bandwidth allocation algorithm of Subsection II-B. With this method we can optimise several sources simultaneously leading to better results.

The summary of the algorithms is as follows. According to the results obtained on the graph model in the source of the demand first, all connections intended for re-routing are torn down in the network. Second, demands are routed by the routing mechanism described in Subsection II-A (or the single-source multicommodity flow subroutine described in this Section) with either the optimistic or the pessimistic strategy. If the routing algorithm is not able to find a feasible route, then the demand request is rejected. Otherwise, a bandwidth is associated to it, and bandwidths of other sources are also set and tuned (adjusted) by iterations as described in Section II-B.

IV. NUMERICAL RESULTS

In this section the performance of the methods is evaluated and compared. The test networks were optimally designed with tightly dimensioned capacities for optimally routed static demands. The input of the design method [15] was the randomly generated position of nodes and a random traffic matrix. The test networks N5, N15, N25 and N35 were of size of 5, 15, 25 and 35 nodes, respectively, also studied and depicted in [16]. Some other characteristic parameters can be seen in Table I. The lower bounds on the demand bandwidths were random numbers between 1 and 9 while the upper bounds were the lower bounds multiplied by a random number between 1 and 4. We assume that in each time slot one demand enters the network with a random duration (holding time) between 1 and 100 time slots.

Networks	No. of Nodes	No. of Links	No. of Commodities
N5	5	5	10
N15	15	15	105
N25	25	31	300
N35	35	51	595

TABLE I
THE 4 EXAMINED NETWORKS

We compared the algorithms on the above four networks from three aspects: the blocking ratio, network utilisation, and running time.

First, different routing strategies have been investigated from the point of view of blocking ratios for all four networks with both, *optimistic* and *pessimistic* allocation strategies for the 5 cost functions as described in Subsection II-A. Two of these ten combinations perform very good on all four networks, namely cost function 5 with optimistic strategy, and cost function 3 with pessimistic strategy. E.g., compared to the simple one-unit weight function blocking of 12.33% was obtained instead of 18.33% for N25; and 5.54% instead of 7.22% for N35). In Figure 1 these strategies are compared with the one unit cost function for N35 (similar results were obtained for the other three networks as well). The network load was increased by linearly increasing both, lower and upper bounds of all demands. Cost function 5 with optimistic strategy yields the best performance, especially in heavily loaded networks.

Figure 2 shows an example how the network utilisation grows during the *Increment Bandwidth Phase* iterations (Subsection II-B). After the *Initialisation Phase* the network utilisation is slightly above 50%. With *Increment Bandwidth Phase*s it will grow: after 10 iterations the network utilisation will be over 95%, while after 20 iterations near 100%. This was the example for N35 consisting of 51 links. The trend is similar in the other three networks as well. Note, that in the worst case the number of iterations is equal to the number of links - 1, however, in our example it was about half of the number of links.

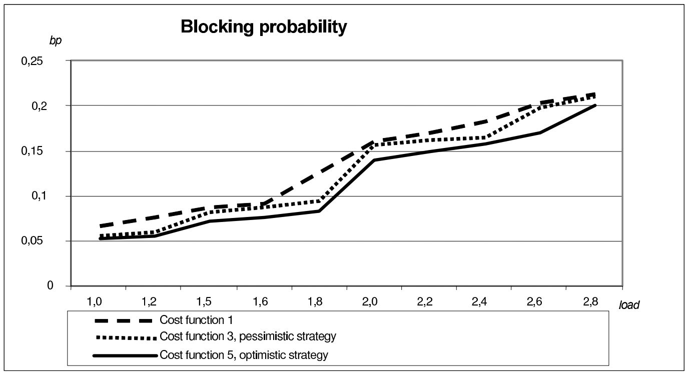

Fig. 1. Blocking ratio against network load ($load$) in N35 for three routing strategies.

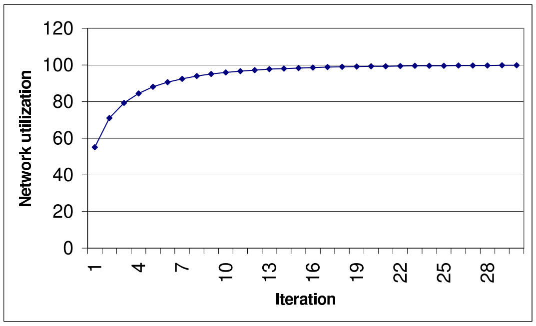

Fig. 2. Network utilisation growth against the number of *Increment Bandwidth Phase* iterations performed.

Next, we study whether it is worth or not using the Iterative Capacity Setting (Subsection II-B). Note, that this method does not affect the blocking probability, but only the throughput. In Figure 3 you can see that the network utilisation is 20% better with any of the cost functions than without iterations, however, the running time has increased by one order of magnitude.

In Figure 4 we compare the two best Dijkstra based allocation strategies, the Shortest Widest Path method, and the Full-Elastic case when the paths can be arbitrarily changed (re-routed). We can conclude that the Shortest Widest Path is slightly better than the Dijkstra function based routing. If the Full-Elastic approach is applied then the blocking probability can be further decreased, especially in less loaded networks (e.g., from 0.2 blocking probability to 0.12).

V. CONCLUSION

We have investigated the on-line routing and bandwidth allocation of working paths paths as well as of restoration paths for elastic traffic. Different routing and resource allocation policies and algorithms have been proposed supported by numerical results. The most important conclusions of this work are:

- The blocking probability can be decreased significantly by using enhanced routing strategies for elastic traffic. Cost function 5 (i.e., link cost is the reciprocal value of the amount of free space on the link) with optimistic strategy proved to be the best one.
- In our approach not a set of alternative paths is available to choose from, but a path is found adaptively among all

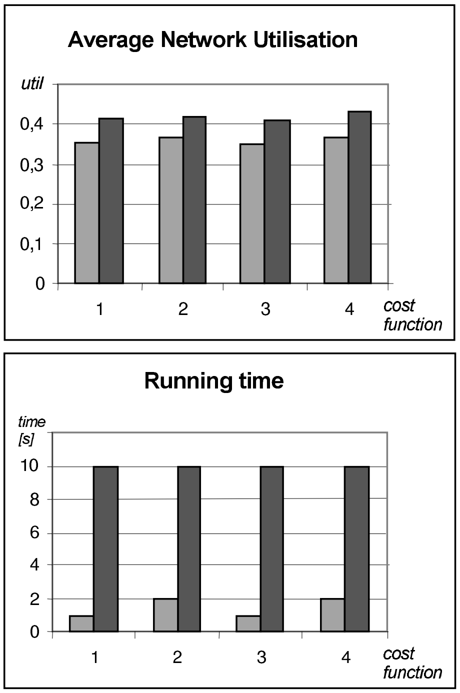

Fig. 3. Network utilisation and running time without (light-grey) and with Iterative Capacity Setting (dark-grey) (Subsection II-B).

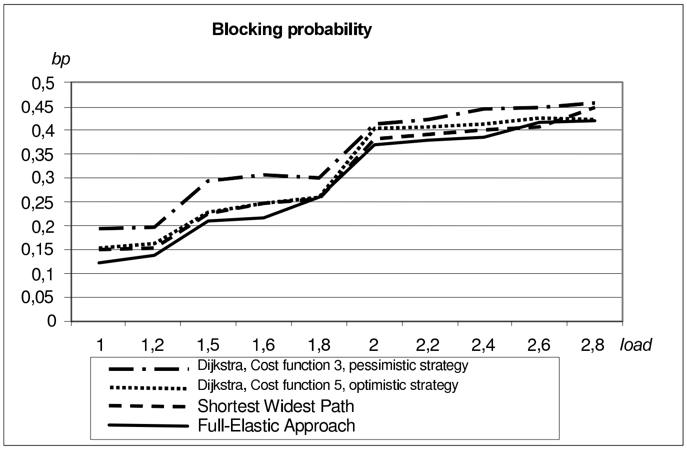

Fig. 4. Blocking ratio against network load in N35 (3 Semi-Elastic cases compared to the Full-Elastic Case).

possible alternatives. Moreover, paths are not determined for new demands only, but it is allowed to re-route some of demands in case of a failure or just to obtain better performance (e.g., the blocking ratio decreased from 0.2 to 0.12 in our case.) Allowing re-routing performs better for lower network loads.

- The proposed method works for on-line routing with a distributed control plane while it obtains results as if it was a centralised method.
- The proposed Iterative Capacity Setting method allows utilising yet spare capacities, without deteriorating the fairness resulting in about 20% higher average network utilisation.
- All the proposed algorithms are a tradeoff (compromise) between blocking probability, network throughput, fairness and computational time.

REFERENCES

[1] A. Riedl, M. Perske, T. Bauschert, A. Probst, *Dimensioning of IP Access Networks with Elastic Traffic,* Networks 2000, Toronto, September 2000.

[2] D. Bertesekas, R. Gallager, *Data Networks,* Prentice Hall, 1987.

[3] A. Charny, D. Clark, R. Jain, *Congestion Control with Explicite Rate Indication*, IEEE ICC '95, June 1995.

[4] F.P. Kelly, A.K. Maulloo, D.K.H. Tan, *Rate Control for Communications Networks: Shadow Prices, Proportional Fairness and Stability,* Journal of the Operational Research Society (49), pp.237-252, 1998.

[5] G. Fodor, G. Malicskó, M. Pióro, T. Szymanski, *Path Optimization for Elastic Traffic under Fairness Constraints*, ITC17, Salvador da Bahia, Brazil, December 2001.

[6] L. Massouline, J.W. Roberts, *Bandwidth Sharing: Objectives and Algorithms*, IEEE INFOCOM'99, 1999.

[7] Y. T. Hou, H. H-Y. Tzeng, S. S. Panwar, *A Simple ABR Switch Algorithm for the Weighted Max-Min Fairness Policy,* IEEE ATM Workshop 97, 1997.

[8] H. Qingyanga, D. W. Petr, *Global Max-Min Fairness Guarantee for ABR Flow Control,* IEEE INFOCOM '98, 1998.

[9] Q. Ma, P. Steenkiste, H. Zhang, *Routing High-bandwidth Traffic in Max-min Fair Share Networks*, SIGCOMM'97, 1997.

[10] T. Cinkler, P. Laborczi, *Joint Routing and Bandwidth Allocation for Protected Elastic Traffic*, 9th International Telecommunication Network Planning Symposium (NETWORKS 2002), München, 23-28 June 2002.

[11] T. Cinkler, P. Laborczi, M. Pioro, G. Malicsko, G. Fodor, A. Szentesi, *Algorithms for Path and Bandwidth Allocation for Elastic Traffic under various Fairness Constraints*, 2nd Polish-German Teletraffic Symposium, September 23-24, 2002, Gdansk, Poland

[12] S.P. Abraham, A. Kumar, *A New Approach for Asynchronous Distributed Rate Control for Elastic Sessions in Integrated Packet Networks*, IEEE/ACM Transactions on Networking, vol.9, No.1, February 2001.

[13] Y. Dinitz, N. Garg, M. X. Goemans, *On the Single-Source Unsplittable Flow Problem*, 39^{th} Annual Symposium on Foundations of Computer Science, November 1998, Palo Alto, California

[14] J. Harmatos, P. Laborczi, *Dynamic Routing and Wavelength Assignment in Survivable WDM Networks*, Photonic Network Communications, pp. 357-376, Vol. 4., No.3/4, July-December 2002.

[15] T. Cinkler, I. Andersson, M. Zethzon, *Greedy Algorithms for Topological Design*, NETWORKS'98, Sorrento, October 1998.

[16] T. Cinkler, P. Laborczi, Á. Horváth, "Protection through Thrifty Configuration", Proceedings of 16th International Teletraffic Congress (ITC16), Edinburgh, pp. 975-988, 7-11 June 1999.