



Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics

Traffic Engineering Algorithms for MPLS Networks

Ph.D. Dissertation Summary

by

Balázs Gábor Józsa

*High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics*

Advisors:

Gábor Magyar, Ph.D.

*Traffic Analysis and Network Performance Laboratory
Ericsson Hungary Ltd.*

Tibor Cinkler, Ph.D.

*High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics*

Budapest, September 2004

1 Introduction

In the dissertation my results are presented on some traffic engineering (TE) problems that were faced during my research work in the field of multiprotocol label switching (MPLS) networks. However, most of the results to be presented are not limited to MPLS environment but they can be applied to other network technologies (e.g., asynchronous transfer mode) as well, where explicit routing and bandwidth reservation can be performed.

Traffic Engineering

Traffic engineering defined by the Internet Engineering Task Force (IETF) [1] deals with the issue of performance evaluation and performance optimization of operational IP networks. The main goal of TE is to enhance the performance of networks using two different approaches. The *traffic oriented* approach of TE addresses the assurance of quality and availability requirements for network users. On the other hand, service providers focus on the economical and reliable utilization of network resources, which is the *resource oriented* approach of TE.

Since the basic purpose of networks is to transfer information from the source to the destination, their most important task is the routing of traffic. Thus, the one of the most significant TE functions is the control and optimization of routing in order to transfer the traffic in the most efficient way. These functions enable the use of sophisticated centralized routing based on the network state, policy, and decision input variables.

TE is not a one time goal but a continual and iterative process of network performance improvement, therefore the TE functions should be used periodically in order to achieve good network performance in the long term. However, the optimization objectives of TE may change over time implied by the new requirements, new technologies, and new problems in the network, which should be taken into account by the TE functions.

MPLS

Multiprotocol label switching (MPLS) [2] is a framework developed for forwarding IP datagrams efficiently in core networks. In MPLS two types of routers are distinguished: (1) label edge routers (LERs) that connect the MPLS network with the classical IP network, and (2) label switching routers (LSRs) that are connected only with MPLS capable routers. The traffic between LERs uses pre-established label switched paths (LSPs) that can be defined explicitly using the traffic engineering extension of resource reservation protocol (RSVP-TE) [3] or constraint based label distribution protocol (CR-LDP) [4]. Moreover, bandwidth can be reserved for each LSP to support TE, which is assumed throughout the dissertation.

MPLS, when combined with traffic engineering, delivers a useful concept for meeting the requirements of differentiated services. A great benefit of this concept is that, it does not need significant changes in the base of the existing IP networks. It fits to asynchronous transfer mode (ATM) switches, which are widespread in the core networks of service providers. In summary, there is a great chance that the next generation routing platform would combine the benefits of MPLS and traffic engineering.

2 Research Objectives

The motivation of my dissertation was to study performance optimization problems in MPLS networks from the point of view of a network operator, furthermore, to propose novel traffic engineering algorithms for this kind of problems.

The issue of optimal resource utilization is coming from the enormous increase in bandwidth requirements caused by novel networking applications as well as by the increasing number of network users. Besides the need for capacity, quality and availability are more and more important factors for network operators in order to be profitable in the competition of service providers.

Traffic engineering with MPLS offers the possibility of making centralized routing decisions enabling the application of sophisticated routing algorithms aiming at improving the performance of operational networks. On the other hand, in the case of planning future MPLS networks, it is advisable to use a design algorithm that takes the ability of route reconfiguration into account resulting in lower deployment costs.

Main Objectives

The main objectives of the dissertation are the following:

- design a robust global path optimization algorithm that can take the specialties of MPLS into consideration,
- improve the backup path calculation and the failure resistance using shared protection,
- propose solutions for the rerouting problem that arises after path optimization,
- compose a multi-hour network design algorithm exploiting the specialties of stepwise cost functions.

Further, the performance analysis of the new algorithms is targeted in order to prove their efficacy.

3 Methodology

During my research work *NP-hard* [5] problems were faced for which—according to the current state of science—an optimal solution cannot be guaranteed in reasonable time for problem instances of practical sizes. As the focus was on the practical applicability, exact algorithms were out of the scope of the dissertation because of their high running time. The exact solution of problem instances involving a few tens of nodes may take several hours [6], while there are also problem instances of practical sizes that cannot be solved due to the huge memory needed. All algorithms to be presented are based on *heuristic optimization* that can offer the required efficiency, moreover the running time can be limited.

There are three basic approaches to evaluate the performance of the algorithms for networking architectures: analysis, simulation, and measurement after prototyping. Analytical methods were widely used for public switched telephone networks (PSTNs) many years ago.

However, some properties of data networks differ from telephone networks, resulting in the inadequacy of analytical methods in most cases. Although prototyping and measurements can generally provide the most accurate results, they were unnecessary for my investigations as the focus was on off-line global optimization problems that are based on computations on centralized data. For these reasons *simulations* were used for performance evaluation.

In all theses the problem of creating many different network topologies arisen in a similar way, therefore a random topology generation method was proposed in Thesis 1. Based on this generator the problem instance generation can be automated, which helped produce a large number of instances improving the reliability of the results. On the other hand, in every thesis real-world topologies were investigated as well, and compared their results to the ones of random topologies.

The network topologies were modeled by directed graphs having capacitated edges. A particular traffic demand was generally determined by its source and destination nodes and its bandwidth requirement. However, in some cases this representation was extended by loose/strict hops, hop limit, and backup paths coming from the traffic engineering capabilities of MPLS networks. In Theses 1–3 the evaluation of novel routing algorithms was based on the *success probability* principal measure indicating the ratio of successfully solved problem instances to the total number of instances. Thesis 4 studies a cost-efficient network design problem, where the *total network cost* was considered as the main performance indicator.

4 New Results

The new scientific results are organized into four main theses with several sub-theses. The corresponding publications and sections of the dissertation are referred in each thesis (e.g., [D3.3] denotes Section 3.3 of the dissertation).

Thesis 1 Global Path Optimization [J1]

The global path optimization (also called global routing optimization) is a basic task of traffic engineering. It has a great practical impact as the changing paths of traffic flows may result in a degraded resource utilization on the long run. This is the case even if dynamic TE is applied enabling by RSVP-TE, because RSVP-TE performs only local optimization. To avoid this, periodical (not too frequent, e.g., daily or weekly) global optimization of paths can be performed by the network operator (or it can be automated as well) while the dynamic TE is disabled. This optimization problem can be transferred to network flow theory in the following way: assign *paths* for a given set of *traffic demands* having capacity requirements in a capacitated graph. This is an unsplittable multiple-source multicommodity flow problem [7, 8] that is NP-hard [5] and has several solutions in the literature. Integer linear programming (ILP) packages can give exact solution, but only for very small problem sizes [9]. For this reason, only approximate solutions [10, 11, 12, 13] and heuristics [14, 15, 16, 17, 18, 19, 20] can be found in the literature. The practical problem sizes are so large (cf. a core network of a next generation mobile telecommunication network) that the running time of the existing solutions in the case of these kinds of problem instances are not acceptable for practical use. Moreover, new technologies (e.g., MPLS) bring novel constraints into the problem (e.g., loose/strict hops, hop limit, and backup paths) that the known methods do not handle. Therefore, it is essential

to design such an algorithm that can solve large problem instances that are equipped with novel technical constraints in reasonable time.

Thesis 1.1 An Efficient Algorithm for Global Path Optimization [D3.3]

The heuristic algorithm proposed for the global path optimization problem (see Figure 1) is based on an iteration that resembles the technique called *simulated allocation* [14, 15]. It maintains two sets: (i) the set of actually unsatisfied demands and (ii) the set of computed paths. In each iteration step, a demand is transferred to a path (*allocation*), or paths are moved back to the set of demands (*deallocation*) for later recalculation. The paths for demands are computed one by one using the well-known shortest path algorithm of Dijkstra [21] in a systematic way. During the iterative path search the possible choices vary with time. The key feature of the algorithm is adapting to the characteristics and difficulties of the particular problem instance by avoiding unnecessary load on bottleneck edges.

The core algorithm involves the following heuristics (indicating the numbers of their corresponding functional blocks in parentheses) in order to explore the difficulties of the actual problem instance enabling the successful solution of a variety of problem instances: demand preprocessing (1), demand difficulty measure (2), insertion of critical demand (2), adaptive edge weight function (3), adaptive path length limit (4), termination condition (5), candidate paths for deallocation (6), edge multiplier adjustment (6), varying number of deallocated paths (6), tournament for deallocation (6), and major deallocation (6). The combination of the above heuristics is the essence of the algorithm and makes it more efficient compared to other methods (including the simulated allocation method). The approach provides enough flexibility in the use of objectives (e.g., maximize the total free capacity, minimize the bottleneck capacity). Furthermore, it facilitates the application of the technical constraints that exist in MPLS networks: loose/strict hops, hop limit, and backup paths.

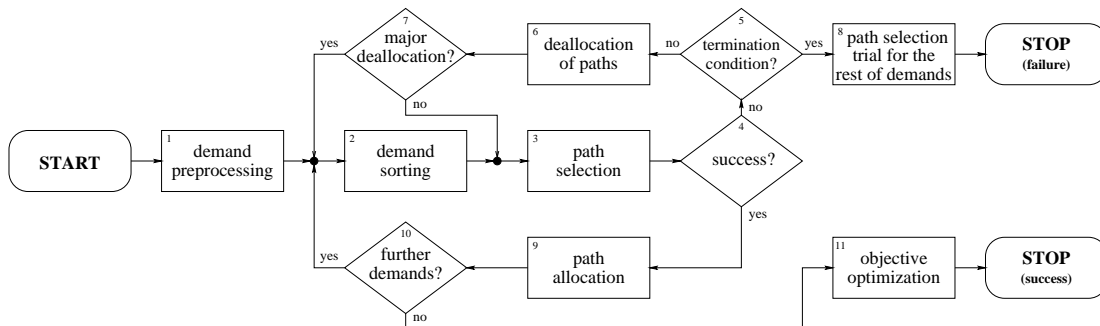


Figure 1: The main functional blocks of the global path optimization algorithm proposed.

Thesis 1.2 Performance Evaluation of the Proposed Global Path Optimization Algorithm [D3.4]

The performance analysis of the proposed global path optimization algorithm was performed by simulation on random as well as real network topologies. In order to generate random topologies a method was developed that imitates the situation in real networks and it

is based on related works in this field [22, 23, 24, 25]. The topology generator has two main input parameters: the number of nodes n and the average nodal degree g . The two parameters immediately give the required number of (undirected) edges that is $\frac{n \cdot g}{2}$. The operation has four main steps. First, the nodes are placed with uniform distribution into a square area that has side length A . The nodes are placed by considering a minimal distance between them, which is determined by $\frac{A}{2 \cdot \sqrt{n}}$. After the location of the nodes has been determined, the graph is made biconnected by adding the edges of a suitable Hamiltonian cycle. The Hamiltonian cycle is calculated starting from the convex hull around the node set and then connecting the remaining nodes by greedy maximization of the connection angle. This cycle yields two disjoint paths between every pair of nodes, which is quite typical in real-world backbone networks. After this, the average degree of nodes is increased by adding new edges into the graph. In each step, the actually longest cycle is divided into two smaller cycles by adding a new edge. Finally, the remaining edges are added iteratively by a probabilistic scheme (see [25]), where the probability of selecting the edge (u, v) is calculated by $p(u, v) = \frac{1}{\mu} \cdot \exp(\frac{-D_{u,v}}{\mu})$ where $D_{u,v}$ is the Euclidean distance between the two nodes, parameter $\mu = \frac{A}{\sqrt{n}}$ is the expected edge length of the exponential distribution. The sampling of the candidate edges for selection is iterated until the required average node degree is reached.

The objectives of the simulation sessions were the examination of the followings: success probability, running time, objective optimization, large problem instances, technical requirements, necessity of the components, strategic constants, and comparison to other methods. Unfortunately, the implementations of other methods were not available, therefore only limited number of comparisons were made (with the help of the developers of the different methods). However, based on these comparisons it can be concluded that the proposed algorithm can produce better results than the other algorithms or it can give results of the same quality in significantly lower time.

The most important result is the ratio of the number of instances for which a feasible solution was found to the total number of instances (namely the success probability) that is shown in Table 1 (each piece of data represents an averaged value of 100 measurements). In the case when the success was not 100%, the average satisfied demand capacity is also included (rounded to the second decimal place) in order to get a more detailed picture of the performance of the algorithm. In the table n refers to the number of nodes in the network and α represents the ratio of extra free resources in the network in the case of routing all traffic demands on minimum hop paths. In the case of $\alpha = 0\%$, when there are no extra free resources in the network, the algorithm could not solve any of the instances disregarding the value of n . This is not a surprising result as the problem is NP-hard so finding a feasible solution is a very difficult task in these “tight” cases. On the other hand, if $\alpha > 3\%$, the success rate is constantly 100%. Furthermore, it can be seen that only the relatively smaller problem instances are difficult, if $n \geq 100$ excellent feasibility results can be obtained even with $\alpha = 1\%$. The reason for this may be that the smaller the instances, the smaller is the number of alternative paths as well as the number of demands, thus the number of feasible solutions is also smaller.

The main results of the performance analysis can be summarized as follows. The proposed algorithm offers advantages compared to familiar methods in terms of running time, flexibility as well as solution quality. It is practically reliable: works for large networks, can handle

n / α	0%	1%	1.5%	2%	3%	5%	10%
25	0/98.21	0/98.91	1/99.16	6/99.53	73/99.94	100	100
50	0/98.79	0/99.60	46/99.89	99/100	100	100	100
75	0/99.27	25/99.85	99/100	100	100	100	100
100	0/99.38	92/100	100	100	100	100	100
200	0/99.50	100	100	100	100	100	100
400	0/99.78	100	100	100	100	100	100

Table 1: Success probability [%] and routed capacity [%] values of the global path optimization algorithm proposed for different problem classes.

specialties of real-world systems, the running times are reasonable, and the results are excellent for feasibility and good for the different optimization objectives.

Thesis 2 Backup Path Calculation Using Shared Protection [C3, C7]

In the current variant of the global path optimization problem the goal is to search for a set of *active* paths and a set of *backup* paths for the set of LSP demands so that the overall bandwidth reservation would be minimal. The MPLS network is represented by a directed graph having edge capacities ($c(e)$ for edge e) denoting the reservable bandwidth of the corresponding link. A traffic demand d_i is given by its source node s_i , target node t_i and by its required bandwidth b_i . The result of the path search is a pair of disjoint paths P_i^1 and P_i^2 per LSP demand d_i , in other words, two ordered sets of edges without common elements. However, the resource utilization when the backup reservations are dedicated to the backup paths (*dedicated protection*) is very low, thus other protection strategies are favored.

The *shared protection* paradigm is preferred by IETF for resource-efficient MPLS-layer protection [26] having the following main idea. Supposing that only one failure can occur at the same time, LSPs having disjoint active paths can share their reservation for the common edges of their backup paths as only one of them can fail simultaneously. This shared backup reservation has been studied extensively in the literature [6, 20, 27, 28, 29, 30, 31] and it seems to be a good compromise between protection and bandwidth consumption. However, the existing solutions do not optimize for the reservation sharing, therefore, it is needed to construct novel approaches aiming at improving the reservation sharing.

The reserved capacity $r(e_j)$ on edge e_j consists of two parts. The first part is reserved for the active paths:

$$\sum_{x:e_j \in P_x^1} b_x. \quad (1)$$

The second part is reserved for the backup paths, i.e., the maximal backup traffic on edge e_i in case of any link failure. This part of reservation can be calculated with the help of the *backup reservation matrix* (BRM) where the rows and the columns denote the edges of the graph. However, assuming that a failure affects a link in both directions simultaneously, BRM has $\frac{m}{2}$ rows in practice, where one row refers to the two edges representing both directions of a failed link. Therefore, pair of edges should be used in the expressions instead of one single

edge but this more complex formulation is disregarded as the results can be transferred to the two-direction case without the impairment of generality. The element $B_{i,j}$ of BRM is the *required backup reservation* for backup paths on edge e_j in case of failure of edge e_i :

$$B_{i,j} := \sum_{x:e_i \in P_x^1 \wedge e_j \in P_x^2} b_x. \quad (2)$$

To sum up, the backup bandwidth reservation (in case of shared reservations) on edge e_j is equal to $\max_i B_{i,j}$ that is the maximal possible backup traffic on edge e_j in case of a link failure.

The task is to search for a set of active paths and a set backup paths for which the total reservation is minimal. It is assumed that active paths are shortest paths, therefore the formal problem is:

$$\text{to minimize } \sum_j \max_i B_{i,j}. \quad (3)$$

Thesis 2.1 New Methods for Efficient Utilization of Shared Protection [D4.3]

Three approaches are presented the purpose of which is to decrease the reservations for backup paths, consequently, to increase the total network throughput. It is a common step of these methods to route a traffic demand having both active and backup paths. This problem is solved by extending the global path optimization algorithm of Thesis 1 by the ability of sharing the reservation of backup paths. The three approaches to be presented require different extensions of the original optimization algorithm as follows.

Cut Down Maximums (CDM). This approach is a postprocessing method, because its starting-point is a complete network configuration (e.g., calculated by [J1]), where both active and backup paths are specified, moreover reservations of backup paths are shared. The main difference between this novel heuristic and the similar published algorithms [27, 28] is that the rerouting of the active paths is allowed as well.

As it was mentioned, the particular allocation for backup paths on a particular edge e_j is determined by the maximum value of column j in the backup reservation matrix (BRM). The idea of CDM is the following: try to cut down the maximum value in a given column while the increase of other maximum values is not significant; in other words the sum of column maximums decreases. Thus, in the case of each successful cutdown the total network load decreases, which is the aim after all. In this way, the set of paths—that are initially not optimized for reservation sharing—can be modified afterwards in such a way that the resulting configuration would have a lower load value.

Adaptive Method (AM). The second approach does not require an initial configuration of paths but it starts from an “empty” network with unsatisfied traffic demands. The main feature of AM is a special, adaptive weight function. When routing the demands one after another BRM changes continuously and the applied weight function is based on the actual values of BRM. Thus, each particular path calculation can be adapted to the current network

situation, i.e., each traffic demand is routed *greedily* resulting in its optimal placement in the actual network configuration.

The main purpose of the adaptive weight function is to keep the column maximums of BRM on a low level. This weight function is basically different for the active and backup paths of a given demand. The two ideas are the following:

- in the case of an active path, the weight of edge e_i is proportional to the average of relative backup reservations required in row i : $w(e_i) \sim 1 + \frac{\sum_j \frac{B_{i,j}}{\max_x B_{x,j}} \cdot c(e_j)}{\sum_j c(e_j)}$, and
- in the case of a backup path, the weight of edge e_j is proportional to the average of relative backup reservations required corresponding to the active path: $w(e_j) \sim \frac{\sum_{i: e_i \in P_x^1} B_{i,j}}{|P_x^1| \cdot \max_i B_{i,j}}$, where d_x is the actual demand.

The proportionality in the above weight functions means that the original weight function of the global path optimizer is multiplied by one of the expressions above depending on whether the calculated path is an active one or a backup one. The aim of the above weight functions is to direct the paths on edges the corresponding BRM element values of which are low, in this way the probability of a new, higher maximum value is reduced significantly.

Iterative Method (IM). The basic idea of IM is to iteratively recalculate all backup paths in the network. In this method two backup reservation matrices (BRMs) are distinguished. B^{actual} contains always $B_{i,j}$ values corresponding the actual backup reservations, while B^{stored} shows a preserved state. Similarly to CDM, this method starts from an initial network configuration for which it stores the actual BRM ($B^{stored} := B^{actual}$) as the first step. Then in each iteration step it performs two operations: (i) it recalculates the backup paths based on B^{stored} with the help of the weight function of AM, which aims to avoid the links with relatively higher reservation values, and (ii) it combines the new BRM values with the previously stored values based on the expression $B^{stored} := W \cdot B^{actual} + (1 - W) \cdot B^{stored}$.

During combination of the two matrices, the relative weight W of the new BRM is decreasing in each iteration cycle. This results in dynamic start of the algorithm, i.e., getting nearer to the optimum quickly. On the other hand, with the help of the increasing ratio of the stored BRM, the oscillation of the reservation values can be avoided.

Combination of the Methods. As the described three methods are based on different approaches, it is a natural idea to combine them in order to get a more effective method. Since AM begins from an empty network, it is practical to start with this procedure. Then, IM or CDM method can also be applied, or the combination of all three methods can be used.

Thesis 2.2 Surviving Multiple Network Failures Using Shared Protection [D5.3]

The one backup path based protection strategy—extensively studied in the literature—was designed for single failures, however, with the help of it multiple failures can also be survived in many cases. Although the backup reservation sharing leads to significantly more

economical capacity utilization, its drawback is that the degree of tolerance against more failures is reduced, too. However, as the size, integration, and complexity of the networks increase, multiple failures [32] become more probable during the operation of the system. Therefore, when serving mission-critical applications—mostly in networks having relatively higher link failure probability—this event has to be taken into consideration as well. The approach of protecting networks against double failures is not unknown in the networking area (see e.g., [33]), however, there is no practical solution in the literature. Therefore, a new method for protecting networks from any two simultaneous failures is introduced, in which three fully disjoint paths are specified for every traffic flow. Moreover, a novel shared protection strategy is proposed, which improves the fault tolerance already in case each traffic flow has only one backup path.

As it was mentioned the task of backup reservation matrix (BRM) is to store the necessary backup capacity values per edge. Preparing for single link failures (when connections have a single backup path) BRM contains $\frac{m}{2}$ rows indicating the possible link failures (assuming that a failure affects a link in both directions simultaneously). This case of shared protection is referred to as ‘SP1’ in the following investigations.

In order to prepare for double failures BRM was modified in such a way that a failure situation i represents the simultaneous failure of two independent links (referred to as ‘SP2’) instead of a single link failure. Thus, the modified BRM has $\frac{(m/2) \cdot (m/2 - 1)}{2}$ rows corresponding to the number of combinations of the two-failure cases. During the comparison of the investigated protection strategies, also the approaches where backup reservations are not shared are examined meaning dedicated protection (referred to as ‘DP’).

A promising idea is to apply SP2 reservation strategy already for the one backup path case, i.e., the network is tried to be prepared to survive double failures. This means that if two LSPs using disjoint active paths have a common link on their backup paths, then the backup reservation on this common link has to be enough to accommodate both traffic flows. However, when more than two disjoint active paths have a common link on the corresponding backup paths, the need for spare capacity can be reduced (compared to the non-shared case). In this way only those LSPs get interrupted the both paths of which are affected by the double failure.

Generalization. In order to prepare for a number of f failures ($f > 2$) in general case, $f + 1$ disjoint paths have to be determined for each LSP demand. The advantage of the above-described novel shared protection strategy is that it can be extended to the case of any number of disjoint paths. However, increasing the number of backup paths, the calculation of paths as well as the sharing of protection capacity become much more difficult. Moreover, during the investigation of real backbone network topologies, it was found that calculating four (or more) disjoint paths between node pairs is impossible in many cases.

Thesis 2.3 Performance Evaluation of the Proposed Shared Protection Methods [D4.4, D5.4]

New Methods for Efficient Utilization of Shared Protection. During the investigation of the performance gain attainable by applying the proposed methods for efficient backup path

calculation, numerous simulation scenarios were performed on various network examples. The networks investigated were generated in the same manner as in Thesis 1.2.

Figure 2 shows the results of the different methods as well as the original global path optimization algorithm (SM) as reference. Combinations of IM are not included in the figure because they could not improve the capacity utilization of the best method in the combination. However, all the other methods can extend considerably the free network capacity, furthermore it was experienced that by combining the methods a higher degree of backup reservation sharing can be reached. Using AM is proposed since it improves network performance nearly to the highest degree, while its running time is at a low level. If time consumption is unimportant, even better network utilization can be reached by applying AM combined with CDM (AM+CDM). The proposed methods AM and AM+CDM were also compared to other published algorithms as well as reference method SM. Based on the performed comparisons, it has turned out that the proposed algorithms give significantly better solutions for the problem.

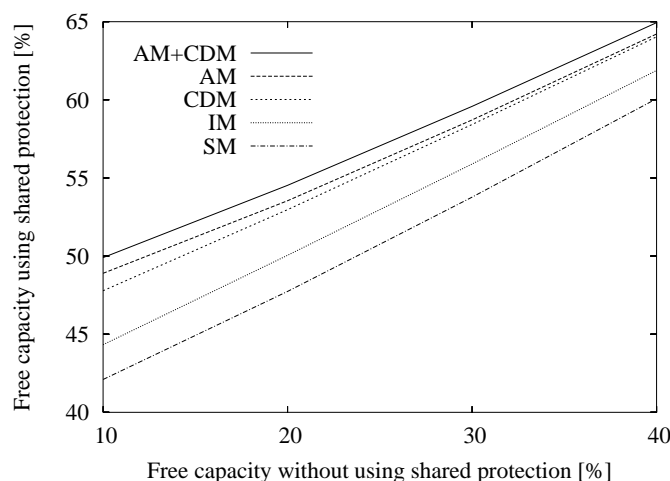


Figure 2: Efficiency of the different backup path calculation methods using shared protection.

Surviving Multiple Network Failures Using Shared Protection. In order to analyze the behavior of the various path protection strategies in multiple failure situations as well as to compare their efficiency thorough simulations were performed. To achieve tri-connected topologies for the investigation of the 2 backup path based protection strategies, after the topology generation a checking procedure is performed in order to verify that 3 disjoint paths exist between every node pair. If this checking fails, then another topology is generated and the checking is performed again.

The main results of these experiments can be found in Table 2 in which each piece of data represents an averaged value of 200 measurements. It shows the average and maximum values of capacity loss (traffic interruption compared to traffic demands) measured in case of 1, 2, and 3 failures as well as the spare capacity reservation in percentage for different network types (number of nodes/average nodal degree). It has turned out that the type of the topology influences the quality of the results significantly. In larger networks the same number of

net type	meas-ure	no backup path	1 backup path			2 backup paths	
			DP	SP1	SP2	DP	SP2
20/ 4.5	+ cap.	0%	136%	53%	83%	306%	119%
	1 failure	4.33/10.1	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0
	2 failures	8.73/18.7	0.14/6.06	0.31/7.82	0.14/6.08	0.0/0.0	0.0/0.0
	3 failures	12.9/26.2	0.42/11.0	0.87/13.8	0.43/11.4	0.018/7.05	0.024/8.10
20/ 3.5	+ cap.	0%	147%	61%	94%	351%	141%
	1 failure	6.83/15.1	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0
	2 failures	13.4/27.1	0.38/11.2	0.63/13.8	0.38/11.7	0.0/0.0	0.0/0.0
	3 failures	19.6/37.1	1.09/20.5	1.73/24.5	1.10/21.1	0.086/13.3	0.100/15.4
50/ 4.5	+ cap.	0%	132%	45%	70%	296%	+103%
	1 failure	2.61/7.49	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0
	2 failures	5.18/14.1	0.05/3.62	0.09/4.97	0.05/3.74	0.0/0.0	0.0/0.0
	3 failures	7.70/20.0	0.14/7.17	0.26/9.23	0.14/7.59	0.004/5.18	0.004/5.66
50/ 3.5	+ cap.	0%	130%	47%	76%	308%	102%
	1 failure	4.44/12.2	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0
	2 failures	8.70/21.3	0.15/7.05	0.25/8.65	0.15/6.84	0.0/0.0	0.0/0.0
	3 failures	12.7/29.5	0.43/13.8	0.70/16.3	0.43/14.0	0.022/6.42	0.024/8.56

Table 2: The spare capacity reservation (+ cap.) needed for the different shared protection strategies and average/maximum capacity loss [%] in case of 1, 2, and 3 failures.

simultaneous failures caused lower capacity loss. The second observation was that the average nodal degree is in inverse proportion to the performance of the capacity sharing. Moreover, also the distribution of the degree of nodes affects the failure tolerance ability, mainly the maximum capacity loss values. Applying the novel shared protection strategy (SP2) for the one backup path case, the reduction in failure tolerance caused by sharing the backup reservations can be avoided. The price of this improvement is about 30% extra capacity compared to the SP1 case, however, the needed capacity is still far below the DP case. Increasing the number of backup paths from 1 to 2, the capacity loss can be eliminated in case of 2 simultaneous failures, moreover also the resistance against 3 failures can be significantly improved, i.e., the average capacity loss values are reduced by more than one order of magnitude in general compared to the one backup path case. On the other hand, the capacity reservation using shared protection remains below the reservation of the 2-path/non shared scheme in most cases.

Thesis 3 Reroute Sequence Planning [C2, C4, J3]

In heavily loaded MPLS networks successive on-demand LSP establishment and deallocation actions may result in a set of LSPs where some paths are not the shortest possible ones, leading to poor resource utilization compared to the optimal state. Thus, global LSP optimization is proposed at certain time intervals (e.g., daily, weekly) to improve network performance, e.g., with the help of the algorithm proposed in Thesis 1. To avoid the interruption of traffic through the LSPs during the rerouting from the old to the new paths, the rerouting of an LSP should be performed using the *make-before-break* concept that involves the following steps. First, the new path of the LSP is established while the traffic is still carried on the old path, then the traffic is switched to the new path, finally the old path is torn down. During this operation (while changing the old paths to the new ones according to a sequence) a problem

may occur, namely, some of the LSPs may not be reroutable to their new paths as there might not be enough bandwidth on some links of these paths, in this way rerouting of these LSPs would cause *capacity violation*. Therefore, such a *reroute sequence* should be planned before the rerouting action is carried out that the rerouting of the LSPs in the calculated sequence would be feasible, i.e., it would not exceed any bandwidth threshold. This reroute sequence planning (RSP) procedure can be realized as a new function of the traffic engineering tool that performs the global path optimization. In [34] it was shown that the reconfiguration of 1000 LSPs takes less than 10 seconds, while the maximal number of LSPs to be rerouted was about 100 after the global path optimization during the preliminary tests. Although the rerouting action takes few time, it must be ensured that the routing configuration does not change during this time for the sake of consistency. Since RSVP-TE has the capability of dynamically reroute the established LSPs [3], this feature should be disabled if RSVP-TE is used for signaling LSPs. The RSP is a new problem in the networking area, thus it has no precedents in the literature.

Thesis 3.1 Problem Formulation of Reroute Sequence Planning [D6.2]

The problem of reroute sequence planning was introduced in [C2], for which now two equivalent formulations are given. This problem is NP-complete, which can be proven by reduction to the well-known *partition* problem.

Graph Based Formulation. Let the MPLS network be represented by the directed graph $G(V, E)$ having the n -element set of vertices V and the m -element set of directed edges $E \subseteq \{(u, v) : u, v \in V, (u \neq v)\}$ corresponding to the nodes and links, respectively. The set of edges is endowed with a non-negative edge capacity function $c : E \rightarrow \mathbb{R}^+$ representing the total reservable bandwidth values of the links. A directed path P with source node s and target node t is defined as a sequence of x edges $\{e_1 = (s, u_1), e_2 = (u_1, u_2), \dots, e_x = (u_{x-1}, t)\}$. Let the structure of an LSP l_i be described by $(s_i, t_i, b_i, P_i, Q_i)$, where P_i and Q_i are the old and new paths, both having source node s_i , target node t_i , and required transmission bandwidth b_i . Note that the presence of LSPs having backup paths does not modify the problem, however, in this case the paths (P and Q) corresponding to the LSPs should contain the edges of both the active paths and the backup paths.

In order to define the problem, the k -element set of LSPs $\mathcal{L} = \{l_i : 1 \leq i \leq k\}$ is given, and L_i is introduced representing the reserved capacities on the edges after the i^{th} rerouting action. It is assumed that the system of the old paths with the corresponding capacities is feasible as well as the system of the new paths, i.e., for each edge the given edge capacity $c(e)$ is not violated by the paths using that edge: $L_0(e) = \sum_{i:e \in P_i} b_i \leq c(e)$ and $L_k(e) = \sum_{i:e \in Q_i} b_i \leq c(e), \forall e \in E$. It is also supposed that $P_i \neq Q_i$ ($1 \leq i \leq k$) as LSPs with unchanged paths can be eliminated and the corresponding edge capacities can be decreased with the result that the equivalent problem without unchanged LSPs is obtained.

The goal in the RSP problem is to determine a reroute sequence (a permutation) $\pi = \{\pi_1, \pi_2, \dots, \pi_k : \pi_i \in \{1, 2, \dots, k\}, i \neq j \Rightarrow \pi_i \neq \pi_j\}$ of LSPs that enables the LSP rerouting without exceeding the capacity constraints, i.e., $\sum_{i:e \in Q_{\pi_i}, i \leq j} b_{\pi_i} + \sum_{i:e \in P_{\pi_i}, i \geq j} b_{\pi_i} \leq c(e), \forall e \notin P_{\pi_j} \cap Q_{\pi_j}, 1 \leq j \leq k$, and $\sum_{i:e \in Q_{\pi_i}, i \leq j} b_{\pi_i} + \sum_{i:e \in P_{\pi_i}, i \geq j} b_{\pi_i} - b_{\pi_j} \leq c(e), \forall e \in P_{\pi_j} \cap Q_{\pi_j}, 1 \leq j \leq k$. As one can see, the capacity constraint expression depends on whether $e \in P_{\pi_j} \cap Q_{\pi_j}$

holds, because the common edges of the old and new paths of an LSP should not be reserved twice during the rerouting [35]. As the solution of the problem, the first LSP to be rerouted will be l_{π_1} , the i^{th} one will be l_{π_i} , and the last one will be l_{π_k} .

Vector Based Formulation. The RSP problem is related to *discrepancy theory* [36] in the following way. Let p_i and q_i be the incidence vectors—both having m elements—of paths P_i and Q_i , respectively ($p_i(e) = 1$ if $e \in P_i$, otherwise $p_i(e) = 0$). Consequently, the rerouting of LSP l_i corresponds to the vector $y_i = b_i \cdot q_i - b_i \cdot p_i$ where each vector component represents the net change of capacity reservations. Therefore, defining the initial vector $L_0(\leq c)$ representing the initial capacity reservations in the network, and the set of vectors $\{y_i : 1 \leq i \leq k\}$, the task is to find a permutation $\pi = \{\pi_1, \pi_2, \dots, \pi_k\}$ of vectors y_i so that their partial sums never exceed c for any vector component, i.e., the edge capacities are not violated at any point during the rerouting process. Formally, find a permutation $\pi = \{\pi_1, \pi_2, \dots, \pi_k\}$ of $1, 2, \dots, k$ so that

$$L_0(e) + \sum_{i=1}^j y_{\pi_i}(e) \leq c(e) \quad \forall e \in E, 1 \leq j \leq k. \quad (4)$$

Thesis 3.2 Algorithms for Solving the Reroute Sequence Planning Problem [D6.4]

Heuristic algorithms are proposed for solving the RSP problem, which are based on an iteration in which one LSP is selected and then rerouted in each step. Note that term ‘rerouting’ in the algorithm descriptions does not mean real LSP reconfiguration in the MPLS network but a modification on the graph representation; the real rerouting action takes place after the termination of the algorithms based on the resulting sequence. An algorithm terminates when all LSPs have been selected, therefore the reroute sequence is built up one by one greedily. The key element is the selection of the LSP to be actually rerouted, which is the base of the difference between the algorithms. It is a common feature of these approaches that the selection is based on reroutable LSPs if there is any, otherwise such an LSP is selected the rerouting of which violates some of the edge capacities, however, the violation is aimed to be kept at a low level. The algorithms assign a *greedy utility* value o_i to each LSP l_i to be rerouted in each step of the iteration, and the chosen LSP is the one that has actually the greatest greedy utility value. The different algorithms assign the greedy utility values o_i for the candidate LSPs as follows.

Random Sorting (RS). RS selects the LSPs randomly: o_i is a random real number taken from the $[0,1)$ interval. This is a very simple method, but it is quick and provides a good benchmark for comparisons, moreover the assignment of o_i can be done in constant time.

Minimal Violation (MV). MV is the simplest greedy method. It calculates for each non-rerouted LSP l_i the greatest capacity violation on its new edges (that are distinct from every old edge) if it is rerouted: $o_i = -\max_{e \in Q_i \setminus P_i} \{b_i + r(e) - c(e)\}$ where $r(e)$ is the actual reserved capacity on edge e . If some edges would be violated in case of rerouting LSP l_i , $o_i < 0$, otherwise $o_i \geq 0$. The idea behind this ranking is to decrease the minimal free capacity on the edges in the slightest degree possible or in case of inevitable capacity excess, to cause a minimal amount of violation of the edge capacities.

Maximal Freeing (MF). MF uses a capacity value $a(e)$ to be routed for each edge e , i.e., the summed bandwidth of such LSPs that has to be allocated to the given edge in the subsequent rerouting steps. Formally, $a(e) = \sum_{j:e \in Q_j \setminus P_j} b_j$ for all j where l_j is not rerouted yet. In this case, the value of o_i represents the total amount of capacity that is to be freed on the old edges of l_i for the subsequent LSP reroutings: $o_i = \sum_{e \in P_i \setminus Q_i} \{a(e) - c(e) + r(e)\}$. The idea of this rule is to prefer those LSPs at the selection which contain edges (in their old paths) that are present in the new paths of many non-rerouted LSPs and have relatively low actual free capacities.

Most Reroutable (MR). In the case of MR o_i is calculated to represent the number of LSPs that can be rerouted without capacity violation after the successful rerouting of LSP l_i . Using this approach after selecting an LSP, in the next step the algorithm can select from the maximal number of reroutable LSPs. In this case more than one LSP may have the same greatest value o_i , consequently the utility value is modified in the following way: the value of o_i is decreased by the summed ratio of the number of non-reroutable edges and the total number of edges to be rerouted for all non-rerouted LSPs after l_i has been rerouted. The important benefit of this approach is that it considers the following rerouting step. On the other hand, it has relatively larger computational complexity compared to the previous algorithms, thus its application is more time consuming.

Enhancements. The enhancements of the above-described greedy algorithms are based on the following fact. The normal RSP problem—rerouting LSPs from paths P to Q —is equivalent to the reverse problem when the task is to reroute every LSP l_i from path Q_i to P_i , and the initial and final capacity reservations are inverted. The reason for this is the following: if there is a feasible sequence for the reverse problem, the reverse sequence is an appropriate solution for the normal problem and this is true inversely. Based on this, three improvements of the above described greedy algorithms are presented resulting in three new variants of each algorithm.

1. The simplest improvement is trying to solve both the normal and the reverse problems. First, the easier problem is considered, which is generally rerouting from that starting state where the total free capacity is less—this is typical in real situations because the goal of path optimization is to increase the total free capacity in the network.
2. In the second variant of the algorithms the sequence is built up from both ends, i.e., LSPs to be rerouted are coming from the solution steps of the normal and reverse problems, alternately.
3. The most complex modification is using backtracking in the algorithms. The algorithm starts and if it gets stuck, i.e., there is no LSP to be rerouted without capacity violations, it deletes a part of the latest inserted LSPs from the sequence (and reroutes the LSPs to their old paths) and continues with the reverse problem. If it gets stuck again it steps back again and changes direction. To avoid infinite loops: at every back-step phase fewer LSPs are deleted from the sequence than the number of LSPs rerouted in the previous phase. In the current implementation the ratio of deleted LSPs from the sequence and

inserted LSPs in the last phase is set to 95% (that value was determined by a fine-tuning process).

Alternative Solutions. During the investigations it has turned out that there are network situations for which feasible solutions cannot be found by the enhancements of the above-described algorithms. In these cases there are two possible approaches to follow: (i) concluding that there is no solution and the LSPs cannot be reconfigured to their new improved paths, or (ii) trying to reconfigure LSPs while allowing some interruption or degradation of traffic during the rerouting process. Following the latter case, first such a reroute sequence is determined that can violate some capacity constraints and then the violations are eliminated by one of the following methods.

- The *interrupting* method allows some LSPs to be interrupted during the rerouting process. After planning the reroute sequence some LSPs are selected so that the summed bandwidth values on their old links cover all the bandwidth excess. The reconfiguration procedure should be started with the deallocation of these selected LSPs and then the rerouting of the LSPs should be performed based on the planned sequence.
- The *shrinking* method decreases the reserved bandwidth of the LSPs during the rerouting process. This results in temporary service degradation but in fortunate cases this amount of degradation is so small that the network users do not perceive it. In this method the reroute sequence planning is the first step again. After this step the bandwidth reservations of the LSPs are decreased so that all violations would be eliminated. In the simplest case the bandwidth values are decreased uniformly: the original values are multiplied by $1/(1+x)$, where x is the maximal violation (defined precisely in the next section). Then the rerouting action follows, finally the bandwidth values are restored.

Thesis 3.3 Performance Evaluation of Reroute Sequence Planning Heuristics [D6.5]

The performance evaluation of the reroute sequence planning algorithms was performed on randomly created traffic situations of real topologies with the help of simulation. Since it was aspired to produce difficult but solvable test instances, after creating an initial network configuration the total capacity of every edge was set to its reserved capacity, i.e., the free capacities of the edges were zero. Then previously defined number of LSPs were reconfigured to new paths one by one so that if the reconfiguration would violate any capacity constraint, the capacity of the edges were increased by the minimally necessary value. In this way “tight” problem instances were generated for which the solutions were available. 1000 problem instances were examined for which the number of nodes, the average nodal degree, and the number of LSPs to be rerouted was taken randomly from the intervals of [10,50], [3,6], and [10,200], respectively. Four metrics were defined to compare the different methods, the calculations of which needed the notation of edge violations during the reroute sequence planning:

- *success probability*: the ratio of the number of cases without capacity violations to the total number of examined test instances,
- *maximal violation*: the greatest edge violation during the rerouting process compared to the total capacity of the edge,

- *edge violation ratio*: the ratio of the number of violated edges to the total number of edges (m),
- *capacity violation ratio*: the summed absolute values of maximal edge capacity violations compared to the total capacity in the network ($\sum_{e \in E} c(e)$).

During the investigations the methods were compared also to the completely Random Order (RO) reference, which differs from RS as it can select from LSPs that are not reroutable without violations also in the case when there are reroutable LSPs.

Consider Table 3, in which the results of the base algorithms (without enhancements) for each metric are shown. It is surprising that the success probability of RO is nearly zero, roughly speaking it could not succeed in solving any of the problem instances. On the other hand, RS—which is a modified version of RO—solved more than the half of the instances. It has also turned out that MR is significantly better than the other greedy approaches in terms of any metric. The reason of this could be the fact that it considers the upcoming rerouting step. Methods RS, MV, and MF gave nearly the same result but surprisingly RS was the best in terms of success probability and edge violation ratio. MV was the best among the three algorithms in terms of maximal violation and capacity violation ratio, and MF was the worst at all. However, it is important to note that these results depend on the instance generation to a great extent and in another context the algorithms might perform differently.

algorithm	success probability	maximal violation	edge violation ratio	capacity violation ratio
RO	0.1	38.29	27.23	1.7224
RS	55.0	3.95	1.12	0.0419
MV	51.6	1.72	1.34	0.0239
MF	50.4	5.93	1.53	0.0572
MR	81.8	1.50	0.37	0.0044

Table 3: Results of the reroute sequence planning base algorithms based on the four defined metrics [%].

Table 4 shows the success probabilities of the different enhanced heuristic algorithms. Some combinations of the enhancements are also included, which means that both enhancements were applied to the problem instances. However, it was unnecessary to combine the first and third enhancements because if the third one cannot solve the problem, the first one cannot solve it either, due to the fact that the third enhancement is an extension of the first one. The second enhancements of the algorithms gave the worst results because for examined tight examples it was better to build up the reroute sequence by solving the easier problem than switching between the normal and reverse ones. The third enhancement improved the probability of success significantly, and there were only a few test instances that could be solved by the second enhancement but not by the third one (compare columns ‘3’ and ‘2+3’).

algorithm	enhancement				
	1	2	3	1+2	2+3
RS	55.0	22.2	76.4	59.0	77.3
MV	51.6	19.7	63.4	53.3	63.5
MF	50.4	26.2	71.1	55.0	71.4
MR	81.8	75.7	92.1	88.3	92.6

Table 4: Success probability values of the different variants of the reroute sequence planning algorithms [%].

Thesis 4 Multi-Hour Network Design [C8, J4]

In the previous theses it was shown that using MPLS-TE the LSPs can be given explicitly and can be reconfigured while the network operates. The use of reconfigurations according to the daily and/or weekly traffic changes is called *capacity management* as recommended by the Telecommunication Standardization Sector of International Telecommunication Union (ITU-T) [37]. An approach of capacity management is multi-hour design that takes the ability of rerouting of LSPs into account already at the network design phase. Using this kind of network design the periodic change of traffic in space as well as in time can be handled efficiently. Thus, the network should not be dimensioned for the case when the traffic volumes are maximal between each source-destination pair but the time scale should be divided into several intervals and different traffic demand sets can be assigned to the intervals. Therefore, the capacity need may be smaller, i.e., money can be saved at the network installation.

Similar problem for ATM networks was already discussed in the literature. In [38] the problem is divided into three sub-problems and during the iterative solution two of them are considered as static at one time. Medhi has several methods [39, 40, 41] for the problem, moreover he also investigated the issue of protection. However, the previous approaches do not use stepwise cost functions and/or they restrict the paths between node pairs to pre-determined path sets. Therefore, it is useful to construct a novel multi-hour network design algorithm that can exploit the specialties of stepwise cost functions and can select optional paths between the node pairs.

Thesis 4.1 A Novel Algorithm for Multi-Hour Core Network Design [D7.3]

A novel algorithm is proposed for multi-hour network design. The proposed multi-hour core network design algorithm CND^{MH} has the same three phases as its single-hour design version CND presented in [C5]. The particular phases are extended according to the more complex problem implied by the more sets of traffic demands corresponding to the different time intervals. The first phase is the initial capacity estimation that estimates the approximate capacity needs. Then the iterative routing optimization follows, the task of which is to provide a feasible network configuration using the output of the first step. The last phase called posterior capacity refinement serves for finalizing the actual network configuration to achieve a cheaper solution. The second phase is the main phase that provides a valid solution for the problem, while the use of the first and third phases is optional, however, it is worth using them as they contribute significantly to the quality of the solution. It is also important that in the

special case when the number of time intervals equals to 1 all phases of CND^{MH} operate in the same way as their single-hour versions.

Initial Capacity Estimation (ICE^{MH}). The goal of ICE^{MH} is to foresee the necessary capacities of network devices by analyzing the sets of traffic demands. This phase provides a partially dimensioned network for the next phase, therefore the remaining tasks are to finish dimensioning of the devices and route the sets of traffic demands corresponding to the different time intervals. Although ICE^{MH} does not give a feasible solution, it specifies a good starting state based on a global view of the problem instance.

The operation of ICE^{MH} is based on the routing of traffic demands one by one in random order. All traffic demand sets are routed a predefined number of rounds using shortest path routing. It is assumed that if all the random routing needed a certain amount of capacity on a particular device, then it is probable that also the optimal routing needs that much capacity. Therefore, the capacity values of the devices are specified in such way that they get the lowest value of the arising ones in the above accommodations.

Iterative Routing Optimization (IRO^{MH}). IRO^{MH} is the main phase of the algorithm, i.e., it can provide a full solution for the problem without the use of the first and/or the third phases. The base of this phase is an algorithm that is capable to route a given set of traffic demands in a capacitated graph. In this paper the algorithm proposed in Thesis 1 is used, which performs well in terms of *feasibility*, namely, it is very probable that it finds a solution for a given problem instance provided that a solution exists. However, the routing optimizer applied can be substituted by any other one.

IRO^{MH} is based on an iteration having two steps. In the first step the traffic demand sets are tried to be routed in the graph while taking the actual capacity constraints into account. If the algorithm terminates with success for all traffic demand sets, then IRO^{MH} terminates. Otherwise, in the second step the capacity of a particular device is increased by one capacity step, and the first step follows again. The selection of the device to be enlarged is done by routing the rest of the traffic demands without capacity constraints for each time interval and then the device having the maximal capacity violation is chosen.

Posterior Capacity Refinement (PCR^{MH}). Although a feasible solution is available after the IRO^{MH} phase, this result can be improved with the help of PCR^{MH} . This phase is based on a local search procedure, i.e., the process concentrates only on one part of the network at one time. The idea behind PCR^{MH} is to reduce the capacity of devices that have a low *relative step utilization* value. The relative step utilization refers to the ratio of necessary and total capacity values on the actual capacity step (that corresponds to the actual device cost).

PCR^{MH} starts with the sorting of the devices by their relative step utilizations—that are averaged for the different time intervals—and then the first device is chosen. The capacity of the actual device is decreased by one step and the sets of traffic demands are tried to be routed by IRO^{MH} in these tighter capacity conditions. If all sets can be routed without cost increase, then PCR^{MH} restarts, otherwise the capacity of the current device is reset and the next device is tried to be shrunk. After the last device is tried to be shrunk without success PCR^{MH} terminates.

Thesis 4.2 Performance Evaluation of the Proposed Network Designer Algorithm [D7.4]

The performance of the novel algorithm was investigated with the help of simulations. Various size random networks as well as real world (a European and a USA) topologies were considered. The main characteristic of the multi-hour design problem is that the bandwidth demand of traffic flows between node pairs is varying in time. In order to create more traffic demand sets a new demand generation procedure was introduced. The network topology was divided into certain number of regions. In a particular time interval each region was matched with exactly one other region. One part of the sum outgoing traffic of a given region specified by parameter Δ flowed towards its dedicated region pair, while the remaining part of traffic was distributed evenly among all regions (including itself as well as its region pair). In this way, $\Delta = 0$ means that there was no extra traffic between the specified region pairs and $\Delta = 100\%$ refers to the case when the total traffic flowed between the specified region pairs. When calculating the traffic demands for a given time interval the original traffic generation method detailed in [C5] was used as first step, then the values for a given region were scaled based on parameter Δ so that the sum of traffic did not change. The difference between time intervals was based on the varying matching of regions.

In order to get a reliable picture, various traffic situations were tested starting from the evenly distributed one ($\Delta = 0\%$) up to the fully polarized one ($\Delta = 100\%$). Two simple algorithms—based on simple extensions of the original single-hour design algorithm CND—were introduced as reference. The traffic bandwidth maximizer algorithm (TBM) designs the network for the maximal arising demands between the node pairs using CND, while the link capacity maximizer algorithm (LCM) designs separate networks for each time interval using CND and then it determines the maximal capacity values from the resulted networks for each device.

Table 5 shows the total network costs (as percentages of the best solutions) as well as the running time values (that were measured on a personal computer of average performance) for the different network sizes/types. Generally, the new proposal outperformed the reference algorithms. As exceptional situations, in the case of evenly distributed traffic combined with larger networks the performance of the new algorithm sometimes fell behind, compared to reference algorithm TBM. However, this does not contradict the expectations since multi-hour design makes sense when significant difference can be observed between the traffic distributions of various time intervals. The running time of algorithms can be considered moderate also in case of larger networks, regarding that they solve off-line network design task. In summary, the proposed multi-hour design algorithm can provide economical solutions with acceptable running times for the examined network situations.

5 Application of the Results

All my research work was performed as part of Ericsson Research and Ericsson Business Unit projects. The algorithms of Theses 1–3 can be used as functions of a complex traffic engineering tool in MPLS networks. The global path optimizer in Thesis 1 and the reroute sequence planning heuristics of Thesis 3 are already implemented in an IP management tool of Ericsson.

network size	algorithm	Δ					time [minute] min-max (avg)
		0%	25%	50%	75%	100%	
15 nodes	CND ^{MH}	100.00	100.00	100.00	100.00	100.00	0-3 (< 1)
	TBM	100.17	107.32	123.10	145.80	152.23	0-1 (< 1)
	LCM	105.75	108.59	113.76	125.29	126.65	0-2 (< 1)
25 nodes	CND ^{MH}	100.67	100.00	100.00	100.00	100.00	1-20 (7)
	TBM	100.00	104.11	119.12	139.09	144.20	1- 8 (2)
	LCM	106.57	106.94	108.61	114.87	118.20	1-37 (6)
35 nodes	CND ^{MH}	104.83	100.00	100.00	100.00	100.00	1-61 (25)
	TBM	100.00	102.62	116.83	134.33	149.62	1-29 (10)
	LCM	108.25	106.80	107.79	112.34	121.88	1-58 (30)
Europe	CND ^{MH}	100.00	100.00	100.00	100.00	100.00	0-1 (< 1)
	TBM	111.42	106.86	120.06	129.85	128.74	0-1 (< 1)
	LCM	126.74	122.54	127.83	128.40	115.19	0-1 (< 1)
USA	CND ^{MH}	100.00	100.00	100.00	100.00	100.00	0-2 (< 1)
	TBM	108.08	109.14	115.12	126.67	134.96	0-1 (< 1)
	LCM	121.11	121.98	123.12	123.69	117.57	0-2 (< 1)

Table 5: Total network cost [%] and running time [minute] results of the proposed network design algorithm.

The novel shared protection approaches in Thesis 2 could be used by network operators in the near future when enabled by network technologies. The multi-hour core network designer in Thesis 4 was initiated within the IKTA-0092/2002 project of the Hungarian Ministry of Education, which is a research co-operation of Ericsson Hungary Ltd., Budapest University of Technology and Economics, and Kovax 95 Ltd. Due to its realistic cost model and good performance, it can be applied to real network design tasks.

References

- [1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, “Overview and Principles of Internet Traffic Engineering,” Internet Engineering Task Force, Request For Comments (Proposed Standard) 3272, May 2002.
- [2] E. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol Label Switching Architecture,” Internet Engineering Task Force, Request For Comments (Proposed Standard) 3031, January 2001.
- [3] D. O. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, “RSVP-TE: Extensions to RSVP for LSP Tunnels,” Internet Engineering Task Force, Request For Comments (Proposed Standard) 3209, December 2001.
- [4] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, and A. Malis, “Constraint-

Based LSP Setup using LDP,” Internet Engineering Task Force, Request For Comments (Proposed Standard) 3212, January 2002.

- [5] M. R. Garey and D.S. Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness,” Freeman, San Francisco, CA, 1979.
- [6] T. Cinkler, P. Laborczi, and Á. Horváth, “Protection through Thrifty Configuration,” Proc. of 16th International Teletraffic Congress (ITC 16), Edinburgh, UK, pp. 975–987, June 1999.
- [7] J. Kleinberg, “Single-source unsplittable flow,” Proc. of 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1996), Burlington, Vermont, USA, pp. 86–95, October 1996.
- [8] S. G. Kolliopoulos and C. Stein, “Improved approximation algorithms for unsplittable flow problems,” Proc. of 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1997), Miami Beach, Florida, USA, pp. 426–435, October 1997.
- [9] S. Van den Bosch, F. Poppe, H. De Neve, and G. H. Petit, “Multi-Objective Traffic Engineering of IP Networks using Label Switched Paths,” Proc. of International Telecommunications Network Strategy and Planning Symposium (Networks 2000), Canada, September 2000.
- [10] T. Leighton, F. Makedon, S. Plotkin, C. Stein, É. Tardos, and S. Tragoudas, “Fast approximation algorithms for multicommodity flow problems,” Proc. of 23th Annual ACM Symposium on Theory of Computing (STOC 1991), New Orleans, Louisiana, USA, pp. 101–111, May 1991.
- [11] S. Plotkin, D. Shmoys, and É Tardos, “Fast approximation algorithms for fractional packing and covering problems,” *Mathematics of Operations Research*, vol. 20, pp. 257–301, 1995.
- [12] N. Garg and J. Könemann, “Faster and simpler algorithms for multicommodity flow and other fractional packing problems,” Proc. 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1998), Palo Alto, California, USA, pp. 300–309, November 1998.
- [13] L. K. Fleischer, “Approximating fractional multicommodity flow independent of the number of commodities,” Proc. of 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1999), New York City, New York, USA, pp. 24–31, October 1999.
- [14] P. Gajowniczek, M. Pióro, and Å. Arvidsson, “VP Reconfiguration through Simulated Allocation,” 13th Nordic Teletraffic Sem., Trondheim, Norway, pp. 251–260, 1996.
- [15] M. Pióro and P. Gajowniczek, “Simulated Allocation: a Suboptimal Solution to the Multicommodity Flow Problems,” 11th UK Teletraffic Symposium, 1994.

- [16] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, "On-line machine scheduling with applications to load balancing and virtual circuit routing," Proc. of 25th Annual ACM Symposium on Theory of Computing (STOC 1993), San Diego, California, USA, pp. 623–631, May 1993.
- [17] P. Aukia, M. Kodialam, P. V. Koppol, T. V. Lakshman, H. Sarin, and B. Suter, "RATES: a server for MPLS traffic engineering," IEEE Network Magazine, vol. 14, no. 2, pp. 34–41, March–April 2000.
- [18] R. A. Guerin, A. Orda, and D. Williams, "QoS Routing Mechanisms and OSPF Extensions," Proc. of IEEE Global Telecommunications Conference (Globecom 1997), Phoenix, Arizona, November 1997.
- [19] S. Plotkin, "Competitive routing of virtual circuits in ATM networks," IEEE Journal Selected Areas in Communications, vol. 13, no. 7, pp. 1128–1136, September 1995.
- [20] M. Kodialam, and T. V. Lakshman, "Minimum interference routing with applications to MPLS traffic engineering," Proc. of Conference on Computer Communications (IEEE Infocom 2000), Piscataway, NJ, USA, pp. 884–893, March 2000.
- [21] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network Flows: Theory, Algorithms, and Applications," Prentice-Hall, New Jersey, 1993.
- [22] E. W. Zegura, K. L. Calvert, and M. J. Donahoo, "A Quantitative Comparison of Graph-Based Models for Internet Topology," IEEE/ACM Transactions on Networking 5(6), pp. 770–783, 1997.
- [23] K. L. Calvert, E. W. Zegura, and M. Doar, "Modeling Internet Topology," IEEE Communications Magazine, June 1997.
- [24] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," Proc. of ACM SIGCOMM '99, Cambridge, United States, pp. 251–262, 1999.
- [25] B. M. Waxman, "Routing of Multipoint Connections," IEEE Journal on Selected Areas in Communications 6(9), pp. 1617–1622, December 1988.
- [26] W. D. Grover, J. Doucette, M. Clouqueur, D. Leung, and D. Stamatelakis, "New Options and Insights for Survivable Transport Networks," IEEE Communications Magazine, vol. 40, no. 1, pp. 34–41, January 2002.
- [27] R. Kawamura, K. Sato, and I. Tokizawa, "Self-Healing ATM Networks Based on Virtual Path Concept," IEEE Journal on Selected Areas in Communications, vol. 12, no. 1, pp. 120–127, 1994.
- [28] Y. Xiong and L. Mason, "Restoration Strategies and Spare Capacity Requirements in Self-Healing ATM Networks," Proc. of Infocom 97, Kobe, Japan, April 1997.
- [29] Y. Zheng, W. D. Grover, and M. H. MacGregor, "Dependence of Network Capacity Requirements on the Allowable Flow Convergence Overloads in ATM Backup VP Restoration," Electronics Letters, vol. 33, no. 5, pp. 362–363, 27th February 1997.

- [30] W. D. Grover and Y. Zheng, “VP-Based ATM Network Design with Controlled Over-Subscription of Restoration Capacity,” Proc. of 1st Int’l Workshop on Design of Reliable Communication Networks (DRCN 1998), Brugge, Belgium, paper O.33, May 1998.
- [31] S. Kini, M. Kodialam, T. V. Lakshman, and C. Villamizar, “Shared backup Label Switched Path restoration,” Internet Engineering Task Force, Internet Draft, November 2000. (Work in Progress)
- [32] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, “Analysis of link failures over an IP backbone,” Proc. of 2nd ACM SIGCOMM Internet Measurement Workshop, Marseilles, France, November 2002.
- [33] D. A. Schupke, A. Autenrieth, and T. Fischer, “Survivability of Multiple Fiber Duct Failures,” Proc. of 3th IEEE International Workshop on Design of Reliable Communication Networks (DRCN 2001), Budapest, Hungary, pp. 213–219, October 2001.
- [34] C. Rossenhövel, “Alcatel Stars in Multiservice Switch Test,” Light Reading, Inc., Technical Report, Section 8, November 2002.
- [35] J. Ash, Y. Lee, P. Ashwood-Smith, B. Jamoussi, D. Fedyk, D. Skalecki, and L. Li, “LSP Modification Using CR-LDP,” Internet Engineering Task Force, Request For Comments (Proposed Standard) 3214, January 2002.
- [36] J. Beck and V. T. Sós, “Discrepancy Theory,” Handbook of Combinatorics, R. Graham, M. Grötschel and L. Lovász eds., Elsevier, pp. 1405–1446, 1995.
- [37] ITU-T, “QoS Routing and Related Traffic Engineering Methods – Capacity Management Methods,” Recommendation E.360.6, CA, 2002.
- [38] T. Bauschert, “Multihour design of multi-hop virtual path based widearea ATM networks,” Proc. of the 15th International Teletraffic Congress (ITC-15), Washington DC, USA, pp. 1019–1029, 1997.
- [39] D. Medhi, “A Unified Approach to Network Survivability for Teletraffic Networks: Models, Algorithms and Analysis,” IEEE Transactions on Communications, vol. 42, pp. 534–548, 1994.
- [40] D. Medhi, “Multi-Hour, Multi-Traffic Class Network Design for VP-based Wide-Area Dynamically Reconfigurable ATM Networks,” Proc. of IEEE INFOCOM’95, Boston, MA, pp. 900–907, 1995.
- [41] D. Medhi, “Some Approaches to Solving a Multi-Hour Broadband Network Capacity Design Problem with Single-Path Routing,” Telecommunication Systems, vol. 13, pp. 269–291, 2000.

Publications

Dissertation

- [D] **B. G. Józsa**, “Traffic Engineering Algorithms for MPLS Networks”, *Budapest University of Technology and Economics*, 2004.

Journal Papers

- [J1] **B. G. Józsa**, Z. Király, G. Magyar, Á. Szentési, “An Efficient Algorithm for Global Path Optimization in MPLS Networks,” *Optimization and Engineering*, vol. 2, no. 3, ISSN: 1389–4420, pp. 321–347, September 2001.
- [J2] D. Orincsay, **B. G. Józsa**, “Távközlési hálózatok költségghatékony tervezése,” *Híradástechnika*, vol. 58, no. 4, pp. 39–45, April 2003. (in Hungarian)
- [J3] **B. G. Józsa**, M. Makai, “On the Solution of Reroute Sequence Planning Problem in MPLS Networks,” *Computer Networks*, vol. 42, no. 2, pp. 199–210, June 2003.
- [J4] L. Tamási, **B. G. Józsa**, D. Orincsay, “Távközlési hálózatok költségghatékony tervezése a forgalomeloszlás periodikus változásainak figyelembevételével,” *Híradástechnika*, vol. 59, no. 9, pp. 32–39, September 2004. (in Hungarian)
- [J5] L. Tamási, D. Orincsay, **B. G. Józsa**, “Cost-Optimal Design of VoIP Networks Using the VPN Concept,” submitted to *Computer Networks*, 2004.

Conference Papers

- [C1] **B. G. Józsa**, D. Orincsay, Á. Magi, B. Szviatovszki, “On the Use of Trunk Reservation in PNNI Routing,” *Proc. of the 3rd International Conference on ATM (ICATM 2000)*, Heidelberg, Germany, pp. 135–139, June 26–29 2000.
- [C2] **B. G. Józsa**, G. Magyar, “Reroute Sequence Planning for Label Switched Paths in Multiprotocol Label Switching Networks,” *Proc. of the 6th IEEE Symposium on Computers and Communications (ISCC 2001)*, Hammamet, Tunisia, pp. 319–325, July 3–5 2001.
- [C3] **B. G. Józsa**, D. Orincsay, “Shared Backup Path Optimization in Telecommunication Networks,” *The 3th IEEE International Workshop on Design of Reliable Communication Networks (DRCN 2001)*, Budapest, Hungary, pp. 251–257, October 7–10 2001.
- [C4] **B. G. Józsa**, “Reroute Sequence Planning for Protected Traffic Flows in GMPLS Networks,” *Proc. of the 2002 IEEE International Conference on Communications (ICC 2002)*, New York City, NY, USA, pp. 2702–2706, April 28 – May 2 2002.
- [C5] **B. G. Józsa**, D. Orincsay, A. Kern, “On the Use of Routing Optimization for Virtual Private Network Design,” *Proc. of the 7th IFIP Working Conference on Optical Network Design & Modelling (ONDM 2003)*, Budapest, Hungary, pp. 865–880, February 3–5 2003.

- [C6] D. Orincsay, **B. G. Józsa**, D. Kósa, L. Tamási, “Recovery Using Precalculated Paths Based Routing in ATM Networks,” *Proc. of the 10th Computer Networks Conference*, Szczyrk, Poland, pp. 119–128, June 10–13 2003.
- [C7] **B. G. Józsa**, D. Orincsay, A. Kern, “Surviving Multiple Network Failures Using Shared Backup Path Protection,” *Proc. of the 8th IEEE Symposium on Computers and Communications (ISCC 2003)*, Kemer-Antalya, Turkey, pp. 1333–1340, June 30 – July 3 2003.
- [C8] **B. G. Józsa**, D. Orincsay, L. Tamási, “Multi-Hour Design of Dynamically Reconfigurable MPLS Networks,” *Proc. of the 3rd International IFIP-TC6 Networking Conference (Networking 2004)*, Athens, Greece, pp. 502–513, May 9–14 2004.
- [C9] D. Orincsay, **B. G. Józsa**, L. Tamási, “Cost-optimal VPN based VoIP network design,” *Proc. of the 11th International Telecommunications Network Strategy and Planning Symposium (Networks 2004)*, Vienna, Austria, pp. 117–122, June 13–16 2004.
- [C10] D. Orincsay, **B. G. Józsa**, L. Tamási, “On the Use of EKB Routing Strategy in PNNI Networks,” *Proc. of the 2004 IEEE International Conference on Communications (ICC 2004)*, Paris, France, pp. 1003–1007, June 20–24 2004.