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

Design and performance analysis of traffic engineering algorithms in telecommunication networks

by

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

Ph.D. dissertation summary

Supervisors:

Áron Szentesi, Ph.D.
Ericsson Hungary Ltd.

Balázs Szviatovszki, Ph.D.
Ericsson Hungary Ltd.

Tibor Cinkler, Ph.D.
High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics

Budapest, 2005

1 Introduction

The dissertation focuses on novel methods of traffic engineering (TE). Generally, TE can be interpreted as the approach of intelligent performance management that aims at utilizing the available resources in the most efficient way. Various underlying technologies enabling the control of routing can serve as background for TE, however, recent proposals—such as multiprotocol label switching (MPLS)—can provide more possibilities. Besides the efficient utilization of existing networks the planning of future networks is also a challenge since the telecommunication world pays attention to economic factors maybe more than ever. Thus, in the dissertation two main issues of TE are addressed: routing optimization and network design.

1.1 Routing optimization

I present my results in two different areas of routing optimization in the first part of the dissertation. Multiprotocol label switching (MPLS) and private network-to-network interface (PNNI) based asynchronous transfer mode (ATM) networks are considered as underlying technologies. Although both technologies have their own properties, they are similar in some important points:

- *link state information* including reservation and load values is available for the routing method actually applied, which enables the calculation of routes in a sophisticated manner aiming at improving the utilization of resources,
- *explicit routing* is available as an option to specify the whole path of a traffic demand by intelligent route calculation algorithms, and
- *bandwidth reservation* can be performed for the traffic demands, thus, per flow requirements on quality can be also taken into account.

Thesis 1 deals with MPLS networks, where the general way to calculate a route for a new traffic demand is the *constrained shortest path first* algorithm (CSPF). Since data backbone networks are assumed here, the label switched path (LSP) requests represent aggregated traffic flows in general. In this way the interarrival and holding time values can be relatively larger, thus, the immediate response (order of millisecond) is not a critical issue, which allows the on-demand calculation of paths using the latest link state information. However, low reaction time (order of second to minute) is a basic requirement, as in all dynamically operating networks. On the contrary, in Thesis 2 such an ATM network is considered that serves frequently arriving virtual circuit (VC) calls. Since immediate response is required here, the routing is based on stored paths that are calculated in advance. The stored paths are recalculated whenever a significant change occurs in the state of links. Thus, similarly to the MPLS environment, the actual network state can also be taken into consideration during the routing decisions.

1.2 Network design

In the second part of the dissertation the area of cost-efficient network design is addressed. The core network design problem examined in Thesis 3 is a general issue, it can arise in the case of various network technologies such as MPLS or ATM. However, the possibility of controlling the traffic flows is essential for foreseeing the capacity need of network elements.

Thesis 4 addresses the more specific design task of voice over Internet protocol (VoIP) networks. Besides the dimensioning of the network devices the so-called VoIP regions have to be also specified, which results in a more complex optimization problem. In the case of both design problems *stepwise* cost functions are applied enabling the realistical modeling of the practical situations. Due to the mathematical complexity of the discussed problems heuristic algorithms are proposed aiming at approaching the optimal solution with acceptable running time.

1.3 Motivation and objectives

New applications appearing in the area of telecommunication have increasing bandwidth requirements in general. Besides the demand upon enormous capacity, quality and availability of services are also important factors that are frequently controlled by strict contracts between the customer and the provider.

Novel technologies such as MPLS can provide a powerful background for traffic engineering strategies, aiming at improving the utilization of the networks. Sophisticated routing and optimization algorithms that are capable of integrating proactivity as well as reactivity are key issues in the case of a quality based service.

Further, new services make it necessary to build new networks in some cases, while intending to minimize the total cost of the configuration to be deployed. In this way, efficient network design algorithms that can handle the specialities of various environments are essential.

In summary, the motivation of my dissertation was to improve the routing and planning decisions made in connection with telecommunication networks by proposing novel traffic engineering algorithms after studying the routing optimization and cost-efficient design problems arising in current technologies. I have defined the main objectives of my dissertation as follows:

- extend the routing capabilities of MPLS by introducing the general concept of prompt partial path optimization and propose new algorithms that can realize the approach,
- compose a new route calculation strategy for precalculated paths based routing in PNNI that can speed up recovery after a network element failure,
- design a core network planning method that can handle stepwise cost functions efficiently,
- propose a design algorithm that can take the objectives of transport network design into account when solving the problem of region specification in VoIP networks, and
- investigate the performance gain achieved by applying the new traffic engineering algorithms.

1.4 Research methodology

Routing optimization tasks imply high mathematical complexity in general. The majority of problems that need decisions affecting the whole network can be categorized as *NP-hard* [GJ79]. In practice this means that, in accordance with the current state of science, finding the optimal solution cannot be guaranteed within acceptable time considering realistic problem instances. Thus, heuristic approaches are favored in this area aiming at producing adequate results while having reasonable running time.

Although an analytical investigation could provide the deepest insight into the behavior of various routing algorithms, it is generally a very complex and difficult task. Considering the examined routing optimization as well as the network design problems it could be hard to find a tractable analytical approach that can be applied to obtain practical results. Thus, the performance evaluation of algorithms of this type is carried out with the help of simulation techniques in general. In the dissertation this approach is followed in order to investigate the improvement that can be reached by applying the new proposals.

Typically, the networks were modeled as directed graphs with capacitated edges. Traffic demands were simulated on call level, thus, the establishment of a traffic demand meant the reservation of the required bandwidth along its actual path, while at termination the corresponding capacity was released. Thesis 1 and Thesis 2 deal with routing optimization, therefore, the total network throughput value was considered as the main long-term performance indicator. Further, the short-term behavior of the system was also focused when investigating failure situations. In Thesis 3 and Thesis 4 network design problems are examined, thus, the total network cost value was chosen as the most important performance measure.

2 New results

Thesis 1: Prompt partial path optimization in MPLS networks [C2, C3, J2]

For shorter term LSP provisioning a local distributed and automated mechanism is needed, that can quickly react upon arriving LSP establishment requests. Generally, the *constrained shortest path first* (CSPF) algorithm implemented in label edge routers (LERs) is used for this purpose. Various CSPF algorithms can be found in the literature [KA00, WC96, AWK+99, SRS01, KL00, MSZ96, SSJ01]. Each proposal aims at optimizing or improving the network performance by concentrating on different tasks, e.g., balancing the load in the case of low and medium network traffic, or minimizing the blocking in the case of network overload.

The common property of CSPF variants is that they route the new LSP demand on arrival and do not change its path later on. Although the routing methods aim at achieving the best network utilization, decisions that seem to be good under the current circumstances can result in a lower network performance on the long term. A well-known solution for this problem is the *global optimization* of LSPs with the help of a centralized off-line network optimization tool [AAF+93, PG94, Plo95, Wan98, AKK+00, FJM+00, JKM+01, Lil03]. This optimization can be performed periodically, when the network state deviates considerably from the optimal state. The complete optimization of all LSPs—depending on the number of paths and size of network—can take relatively long time on a dedicated server [Lil03]. Furthermore, this action often causes the change of almost all LSP routes, which has two undesirable effects: (1) increased signaling overhead compared to the normal operation, and (2) a transitional state—due to the major path restructuring—in which some LSPs may have to be torn down before their new routes are established [JM03]. For these reasons it is not advisable to perform global optimization too frequently, namely, it should be considered on a larger time-scale, e.g., few days or a week.

The abovementioned methods represent two options in the space of possible routing approaches. Obviously, CSPF is the simplest since it does not affect the paths of already established LSPs, while global optimization—by possibly rerouting all LSPs in the network—is the most complex. However, beyond these approaches there are other options as well. One could intend to introduce a constraint to a global optimization algorithm, and restrict the optimization to a predefined maximum number of LSPs. Alternatively, CSPF could be enhanced as well, by allowing the rerouting of a few already established LSPs. This extension would be used in the case of a CSPF failure, namely, when CSPF does not find a feasible path due to the shortage of reservable bandwidth in the network. Surveying the literature for such algorithms it turned out that these research directions are yet unexplored.

In this Thesis several algorithms are proposed for the latter case, namely, to trigger a prompt partial path optimization (PPPO) in order to route the new LSP when CSPF failure occurs. In this method, instead of rerouting all LSPs of the network, the number of LSPs affected is intended to be minimized. As a consequence of PPPO, reservable capacity may be released along the possible routes of the new LSP demand so that it can be established. Since this action can be performed relatively often, a fast optimization algorithm is necessary that affects only a few LSPs established in the network.

Thesis 1.1: General concept of prompt partial path optimization

I have proposed the general concept of *prompt partial path optimization* (PPPO), whose task is to improve the routing capability of MPLS by allowing to reroute several already established LSPs in order to fit the new LSP demand into the network.

As mentioned before, PPPO is triggered by the failure of the CSPF algorithm that occurs because of the lack of reservable bandwidth between the ingress and egress router of the new LSP demand. The basic process of PPPO can be seen in Figure 1. Note that the upper limit on the number of reroutable LSPs n_{max} can be adjusted, enabling the control of the running time of the algorithm in a sense. In the case of a real-time application a more suitable implementation would be to impose a time limit on the algorithm that is being continuously examined. In this way, when the actual running time exceeds this limit the algorithm stops with failure.

Step 1: $n = 1$.

Step 2: Try to route the new LSP request by rerouting n previously established LSPs.

Step 3: If a feasible solution is found, stop with success.

Step 4: If $n = n_{max}$, stop with failure.

Step 5: $n \leftarrow n + 1$. Go to step 2.

Figure 1: Basic process of PPPO.

The core of the algorithm (see step 2) is a method that is able to reroute one or more previously established LSPs in order to route the new LSP demand. Two different methods are proposed that realize this function. The first one called *integer linear programming based simultaneous rerouting* (ILP-SR) formulates the task of path selection for the new demand and the LSPs to be rerouted as an *integer linear program* (ILP) (see details in Thesis 1.2). The main idea behind the second method called *Dijkstra's algorithm based* *recursive rerouting* (DA-RR) is that it tries to reroute LSPs one by one with the help of Dijkstra's shortest path algorithm (see Thesis 1.3).

Thesis 1.2: Integer linear programming based simultaneous rerouting

I have proposed an algorithm called *integer linear programming based simultaneous rerouting* (ILP-SR) that realizes the core function of PPPO. Figure 2 presents the main steps of the method.

Step 1: Select the first LSP subset (containing n LSPs).

Step 2: If the selected LSP subset can be filtered, go to step 5.

Step 3: Try to route the new LSP demand and reroute the selected LSP subset with the help of ILP.

Step 4: If a feasible solution is found, stop with success.

Step 5: If there are no more LSP subsets, stop with failure.

Step 6: Select the next LSP subset (containing n LSPs). Go to step 2.

Figure 2: Main steps of ILP-SR.

The essence of the above method is step 3, where the routing problem of a given set of LSPs is formulated as an ILP and tried to be solved. As an important characteristic of this approach, the possible paths are sought simultaneously for the LSPs. Generally, any ILP software package can be applied to solve this program, during the simulations I used the *lp_solve* software package [BD97].

The ILP formulation and solution (step 3) is the most time-consuming step of ILP-SR, thus, decreasing the number of its executions could result in a significant improvement in overall running time. For this purpose so-called *filters* are applied in step 2. The feature of these filters is that their running time is negligible compared to the ILP formulation and solution, while using them many LSP subsets can be ignored during optimization, which results in considerable running time reduction.

I have proposed two filters for this purpose. The first one called *routability filter* (RoF) is a *non-heuristic filter*, which means that it examines necessary conditions and consequently precludes only such situations that are surely impossible to solve. It checks whether the new LSP demand could be routed if a given LSP subset is torn down. It is possible that the elements of the LSP subset selected to be rerouted are far from each other in the network. This is the main idea behind the second filter called *relationship filter* (ReF) that tries to make the relationship between LSPs measurable. Based on this measure, the examination of such subsets where one or more LSPs are not in relationship with the others can be avoided. ReF is a *heuristic filter* as it checks a condition that is probably fulfilled if the examined case has a suitable solution, which means that it sometimes also precludes situations that would be solvable.

Although an important feature of the basic ILP-SR is that it surely finds a feasible solution if one exists (for a given LSP subset), by applying heuristic filtering this characteristic gets reduced. However, during the numerical investigations it turned out that heuristic filtering does not result in relevant performance degradation, while the caused running time decrease is significant.

Thesis 1.3: Dijkstra's algorithm based recursive rerouting

I have proposed another new algorithm that is capable of solving the rerouting of more LSPs as core function of prompt partial path optimization concept, namely the *Dijkstra's algorithm based recursive rerouting* (DA-RR) algorithm. The main difference between the proposals is that while ILP-SR routes the LSPs simultaneously, DA-RR selects a path only for one LSP at a time. The principal idea behind this method is the following: (1) find an LSP whose teardown releases the sufficient amount of bandwidth for setting up the new LSP demand, (2) route the new LSP demand, then (3) consider the torn down LSP as a new LSP demand, and (4) try to route this 'new' LSP demand in the same way.

```
procedure darr($lsp_i, l$) {
  if $l = n$ {
    if dijkstra($lsp_i$) {
      establish($lsp_i$)
      return success
    }
    else return failure
  }
  else foreach $lsp_j, j \in \{1, ..., i-1, i+1, ..., p\}$ {
    teardown($lsp_j$)
    if dijkstra($lsp_i$) {
      establish($lsp_i$)
      if darr($lsp_j, l+1$) return success
    }
    else establish($lsp_j$)
  }
  return failure
}
```

Figure 3: Dijkstra's algorithm based recursive rerouting.

This idea can be implemented with a help of a recursive procedure. The pseudo code of this procedure called 'darr' can be seen in Figure 3. As mentioned before, the need for PPPO is indicated by the failure of CSPF resulting from the lack of reservable bandwidth. Moreover, resulting from the basic structure of the algorithm, on a particular optimization level n, it is sure that the new LSP demand cannot be set up by rerouting $n-1$ or less LSPs (see steps of PPPO). Consequently, when the level of recursion l reaches the current optimization level n, the procedure tries to route the actual LSP demand lsp_i without allowing the rerouting of any further LSPs. If level n has not been reached yet, the procedure tears down the previously established LSPs in a cycle, and checks if the actual demand lsp_i became routable. If the actual LSP demand is successfully routed, the procedure is called recursively by considering the LSP torn down as a new LSP demand as well as increasing the number of rerouted LSPs referred to as l. If the procedure does not find an LSP that both enables the setup of the actual LSP demand and can be rerouted itself, it returns to the lower level (in the recursion) with failure. The function 'dijkstra' in the procedure performs a bandwidth constrained path selection by pruning the links with unsufficient amount of bandwidth. It returns with 'success' if the path selection was successful, and with 'failure' otherwise. The function 'establish' reserves the required bandwidth along the path previously selected by function 'dijkstra', while the function 'teardown' releases the corresponding bandwidth.

DA-RR implicitly guarantees that the LSPs to be rerouted are in close relation in all examined cases, since the necessary condition of the origination of a new branch in recursion is that the teardown of the new LSP to be rerouted has to enable the setting up of the current LSP demand. In this way, every examined situation includes the possibility of a successful optimization. Therefore, no further filters are used for DA-RR.

Thesis 1.4: Performance evaluation of prompt partial path optimization

I have performed various simulation experiments in order to investigate the attainable performance improvement resulting from the application of prompt partial path optimization.

The simulations were carried out in a dynamic environment, where new LSP establishment demands are arising continuously. The topologies were generated with the help of the *random topology generator* method presented in [JO99, JKM+01]. In order to investigate different network configurations, the number of nodes varied from 10 to 50, and the average nodal degree was shifted from 2.5 to 4. The links were generally 2.5 Gbps (STM-16) connections. The arrival of new LSP demands was modeled as a Poisson process, while the holding time values were specified by the Weibull distribution. The bandwidth values were generated randomly using a uniform distribution, and the total volume of bandwidth requested in the network was adjusted by changing the expected value of the corresponding random variable. Operators generally endeavor to manage as much traffic as possible, therefore the *total network throughput* was identified as the main performance metric in the numerical comparisons. Further, practical running time is also an important factor from the point of view of applicability.

Running time. In this scenario the practical running time values are analyzed in the cases of different PPPO algorithms and various network sizes. As it can be seen in Table 1, two kinds of *worst cases* were differentiated: (1) the new LSP demand can be established only by rerouting exactly the maximum number of reroutable LSPs, or (2) the new LSP demand cannot be set up at all. The simulations were carried out on a Sun Ultra Enterprise 420R machine with an Ultra II 450 Mhz processor and 1 GByte of memory.

The first observation is that there is a difference of an order of magnitude between the running times of DA-RR and ILP-SR methods in general. It can also be noticed that if DA-RR is allowed to reroute only a single LSP (referred to as DA-RR-1), it needs minimal running time even in the case of a 50-node network, where the ILP based method finishes in a minute on average. Considering 10 minutes as a realistic time limit, it can be concluded that allowing two LSPs to be rerouted, DA-RR can be applied up to 30 nodes. Further, ILP-SR-2 can be used up to network sizes of 20 nodes. The necessary running time of DA-RR-3 in the case of blocking is unacceptably high for a 20-node network, the algorithm reaches the limit of its applicability at about 17-18 nodes. Unfortunately, rerouting three LSPs with ILP-SR may result in almost one hour blocking time (for 10-node networks). Thus, it proved to be unfeasible for real time situations. However, it might be used in

Algorithm	Network size [node]				
	10	15	20	30	50
CSPF	<0.1	<0.1	<0.1	<0.1	<0.1
DA-RR-1	<0.1	<0.1	<0.1	0.6	5.5
	<0.1	<0.1	<0.1	0.6	5.3
ILP-SR-1	<0.1	<0.1	1.8	12.4	52.2
	<0.1	<0.1	2.1	10.3	62.3
DA-RR-2	1.1	8.3	34.8	554.5	n.a.
	0.7	6.7	45.7	556.0	n.a.
ILP-SR-2	8.2	42.2	389.4	n.a.	n.a.
	13.8	87.1	588.2	n.a.	n.a.
DA-RR-3	10.3	58.4	n.a.	n.a.	n.a.
	7.8	44.2	n.a.	n.a.	n.a.
ILP-SR-3	523.4	n.a.	n.a.	n.a.	n.a.
	2751.3	n.a.	n.a.	n.a.	n.a.

Table 1: Running time of algorithms [s].

extremely small networks, or if the number of LSPs is low. Further, applying computers having relevantly higher computing capacities combined with efficient ILP solving methods, the network size that can be handled may increase.

Normal operation with symmetric traffic pattern. In this simulation scenario a network with roughly symmetric traffic was considered, namely the intensities of new LSP requests were nearly the same between the various nodes. The average network load was adjusted by changing the expected value of LSP bandwidth requests. The simulation was performed on a 20-node random network topology having an average nodal degree of 4.

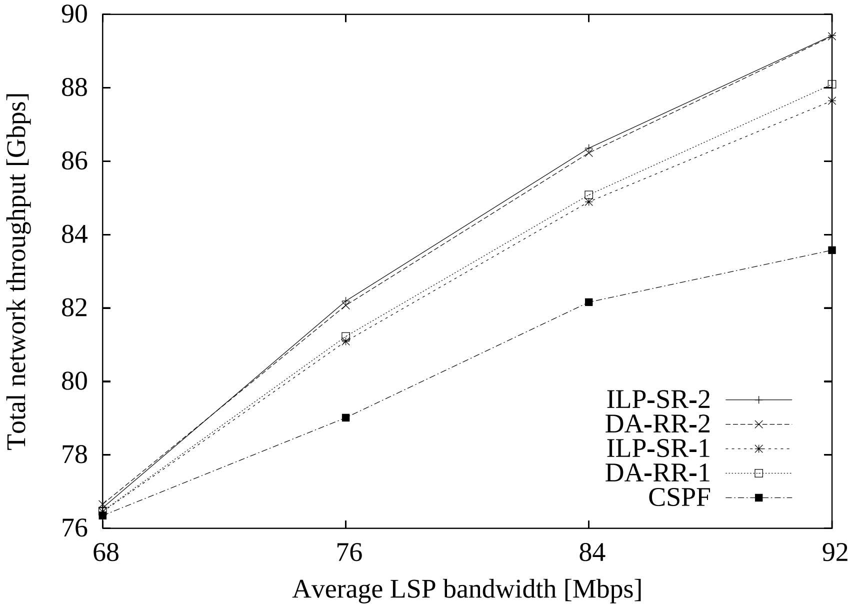

Figure 4: Evenly distributed LSP requests.

As it can be seen in Figure 4, the difference between the algorithms is minimal in the case of average LSP bandwidth of 68 Mbps. This is not surprising since the blocking probability of a new demand is very low at this point. It is clearly visible that increasing the load the benefit from the use of PPPO increases as well. At about 84 Mbps average LSP bandwidth, the performance improvement gainable by rerouting a single LSP is about 4%. Moreover, allowing the rerouting of two LSPs, the attainable gain in total network throughput reaches 6%. By applying ILP-SR as the core function of PPPO, even higher network utilization can be achieved in some cases, however, the average improvement is similar.

Increased traffic volume in a particular node. In order to get a deeper insight into the behavior of the various optimization algorithms, it is worth testing them in special situations. In this simulation scenario such a case was investigated when the incoming and outgoing traffic volume of a particular network node increases unexpectedly due to some special event in the access area of the node, e.g., an earthquake. A 13-node random network having 3.54 links per node on average was examined.

After the stationary state—in order to simulate a special event—the LSP arrival intensities were increased by 200% from and towards a certain node in the 40th time unit, while the intensities between other LERs remained the original. As it can be observed in Figure 5, this change results in a significant growth in the overall throughput. However, the blocking probability increases as well. By applying PPPO a part of the demands blocked by CSPF can be routed, consequently, much higher network throughput can be achieved.

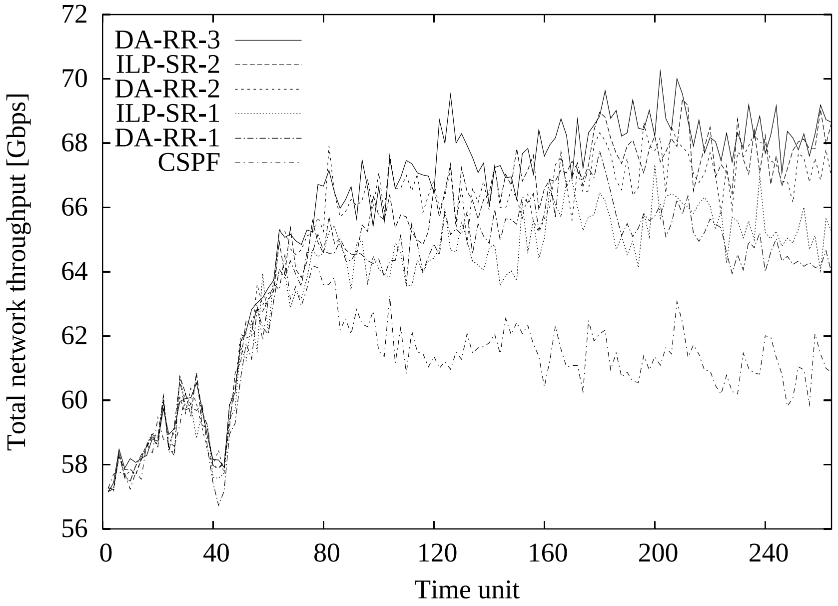

Figure 5: Effect of a special event which occurred in the 40th time unit.

In Table 2 the average throughput values are shown concerning the [80, 260] time interval. When the rerouting of a single LSP is allowed, about 6% improvement can be reached. By rerouting an additional LSP the performance gain increases by a further 3%, resulting in an average throughput of 109% compared to the pure CSPF method. Allowing the rerouting of 3 LSPs in DA-RR the throughput increases again by 1.5%.

Measure	Algorithm					
	CSPF	DA-RR-1	ILP-SR-1	DA-RR-2	ILP-SR-2	DA-RR-3
Throughput [Gbps]	61.52	65.26	65.12	66.83	66.91	67.83
Rel. throughput [%]	100.0	106.07	105.85	108.64	108.75	110.25

Table 2: Average network throughput in the time interval [80, 260].

Thesis 2: Precalculated paths based routing in PNNI networks [C6, C10]

The asynchronous transfer mode (ATM) technology serves as basis for many communication networks. The private network-to-network interface (PNNI) [Atm02] generally used in ATM networks does not specify the particular routing method, thus, various sophisticated routing strategies (e.g., [MSS00, C1]) can be applied. Using up-to-date information about the state of the network as well as smart routing methods is essential to fulfill the ever stricter requirements of customers, which generally include low reaction time upon new call arrival, and high level restoration combined with fast convergence in the case of a network element failure.

A progressive concept is to store a number of precalculated paths for each destination in switches [Top98]. Applying this approach the probability of successful call setup can be improved. Moreover, alternative paths that are available immediately can also be useful in the case of a failure. The common method used for this purpose is the *K shortest paths algorithm* (KSP) [Yen71], which calculates the K cheapest paths (per node-pair) based on an appropriate cost function. Applying link weight functions that can adapt to the network state changes, KSP can utilize the available resources well. On the other hand, regarding possible network element failures KSP has a property that may be a weakness in some cases. Although KSP provides more alternative paths, it does not ensure the existence of a bypass path when a device fails along the current path.

In order to eliminate this characteristic of KSP while keeping its good abilities considering the faultless periods of the operation of network a new routing approach is examined. The main purpose of the novel concept is to reduce the restoration time in case a network device fails. The motivation for this new concept is the relative frequent occurrence of network element failures observed by network operators [ICM+02, NSB+02]. Although some applications—that are typically categorized as best effort (BE)—do not require efforts to hide the temporary disturbances of the communication channel, an increasing number of services need the smooth recovery of the connection in the case of a failure.

Thesis 2.1: Edmonds–Karp algorithm based routing strategy

I have proposed a novel route calculation strategy that improves the availability of networks using precalculated paths based routing by reducing the recovery time of soft-permanent virtual connections (SPVCs) dropped due to a network element failure.

The task of the original Edmonds–Karp algorithm (EK) [EK72] is to find K (two or more) disjoint paths between a given node-pair in such a way that their sum cost would be minimal according to a certain cost function. Although the problem of finding disjoint paths between a node-pair was first solved by Ford and Fulkerson [FF56], the algorithm of Edmonds and Karp applies an improvement that reduces the mathematical complexity resulting in faster operation in practice. Specifying three or more disjoint paths for each node-pair in a topology of a common communication network is unreasonable and typically impossible. Moreover, the purpose of the new concept is to ensure a bypass path when a network element fails along the current path, which generally does not require fully disjoint paths.

The main idea behind the novel proposal, the Edmonds–Karp algorithm based routing strategy (EKB) is that the original EK algorithm is applied to a specially extended version of the original graph representing the ATM network. This modified graph is constructed in the following way. Let k denote the number of paths to be stored between a certain node-pair. For $k \geq 3$ the edges of the original graph are multiplied by $(k-1)$. In this way, $k-1$ parallel virtual links become available between the neighboring nodes, e.g., for $k=3$ the edges of the original graph are doubled. The edge copies have exponentially increasing weights, namely the cost of the i^{th} virtual link is 2^{i-1} multiplied by the actual cost value of the original link.

After the graph extension EK is executed for every node-pair to find k disjoint paths. Since $k-1$ parallel virtual links—representing the original one—coexist for each neighboring node-pair in the extended topology, at most $k-1$ of the calculated paths between a source-destination node-pair use a certain original link. Consequently, at least one of the paths exclude the given link. This means that in the case of a link failure the existence of a bypass path is guaranteed among the stored ones for each connection. Because of the exponentially increasing weights of virtual links it can be avoided that a given (original) link is included unnecessarily in more paths between a certain node-pair.

For $k=2$ the original graph can be considered and the algorithm of Suurballe [ST84] can be used that is a faster algorithm than EK for finding two disjoint paths. Moreover, if $k=1$ Dijkstra's algorithm can be applied directly, however, in this case the immediate recovery is impossible. Since EK and Suurballe's algorithm do not deal with the order of paths found, they are ordered increasingly by the applied cost function.

Thesis 2.2: Performance evaluation of the Edmonds–Karp algorithm based routing strategy

I have carried out numerous simulations in order to analyze the behavior of EKB. During the simulations two main issues were addressed: (1) normal operation of the network, when no failure occurs for a long time period, and (2) failure situation, when a link of the network is cut causing the transient degradation of services.

In the simulations the network topologies were generated randomly [JO99, JKM+01]. Networks having 10 to 30 nodes were examined, however, results corresponding to the 20-node scenarios are detailed here. The links were generally type STM-4 (622 Mbps), having administrative weight of 5040 [Atm02].

SPVC requests were generated randomly with a uniform distribution regarding the source-destination node-pairs. Their bandwidth requirements were uniformly 2 Mbps (type E1 connection). Since SPVCs are typically long-lived, their holding time values were considered as infinite. SVC requests were also generated having a bandwidth demand of 64 kbps and an average holding time of 120 seconds using an exponentially distributed random variable (uncompressed voice call). The arrival of SPVCs and SVCs were modeled as a Poisson process.

Both the novel proposal EKB and the reference algorithm KSP enable the use of different link cost functions, thus, three variants of *link load based* (LB) cost functions (see [C10]) as well as the static *administrative weight* (AW) based one were tried.

Normal operation. This first scenario addresses the investigation of EKB in the case of normal network circumstances, when no failure occurs for a longer time period. Comparing the measured network throughput values presented in Table 3 it can be concluded that relevant difference cannot be realized when using the various routing strategies. However, two tendencies can be observed. First, EKB always overperforms KSP for a given number of stored paths k and a certain link cost function. Second, it is advisable to use load based link cost function, moreover, LB-1 seems to be the most efficient among the examined ones.

Routing strategy	Offered network traffic [Gbps] 4	4.5	5	5.5	6
EKB LB-1	4.00	4.49	4.91	5.26	5.54
KSP LB-1	4.00	4.48	4.89	5.25	5.51
EKB LB-2	4.00	4.49	4.90	5.24	5.49
KSP LB-2	4.00	4.45	4.83	5.14	5.43
EKB LB-3	4.00	4.48	4.90	5.24	5.50
KSP LB-3	3.99	4.44	4.86	5.19	5.46
EKB AW	3.98	4.43	4.85	5.20	5.47
KSP AW	3.94	4.39	4.79	5.13	5.43

Table 3: Total network throughput in normal operation for $k = 3$.

Failure situation. This test scenario aimed at examining the process of rerouting when applying the various routing strategies. As it can be seen in Figure 6, EKB and KSP have similar rerouting probability values when using LB-1 as the cost function. On the other hand, in the cases of LB-2 and LB-3 the use of EKB shows significant gain compared to KSP. Although using AW in EKB provides quite good results at some load levels, it turned out again that it is worth applying link load based cost functions. For this reason, the examination of AW is not addressed in the remaining experiments.

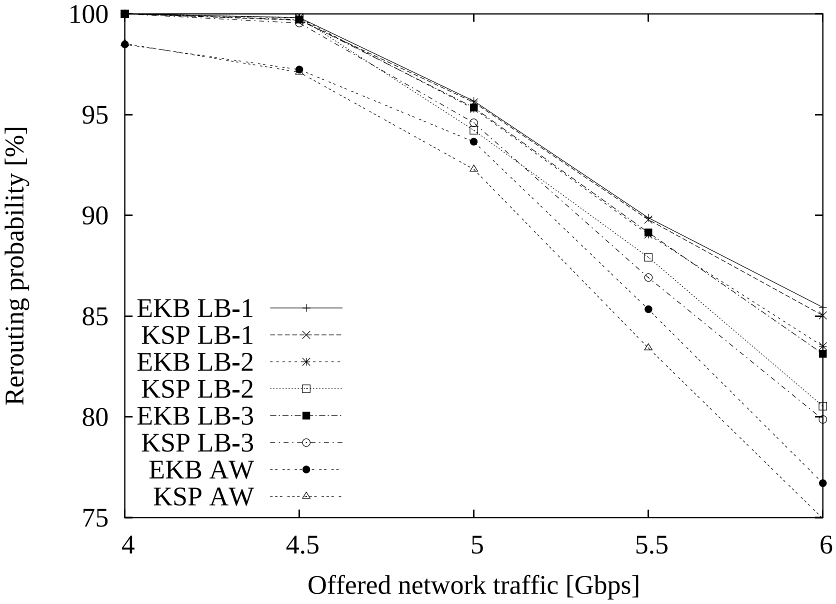

Figure 6: Rerouting probability values for $k = 3$.

As the most relevant performance indicator regarding the current investigation, the average time needed to reroute the SPVCs after link failure was also evaluated. The first observation (see Figure 7) is that increasing the offered network traffic the rise in average rerouting time is quite moderate when using EKB compared to KSP. The most important result is that EKB needs significantly less time to reroute the connections than KSP. Comparing the various link load based cost functions, LB-2 and LB-3 seem to overperform LB-1. One reason for this phenomenon could be that the rerouting capabilities of them are lower, thus, the calculation of average rerouting time value is based on a reduced set of connections in these cases.

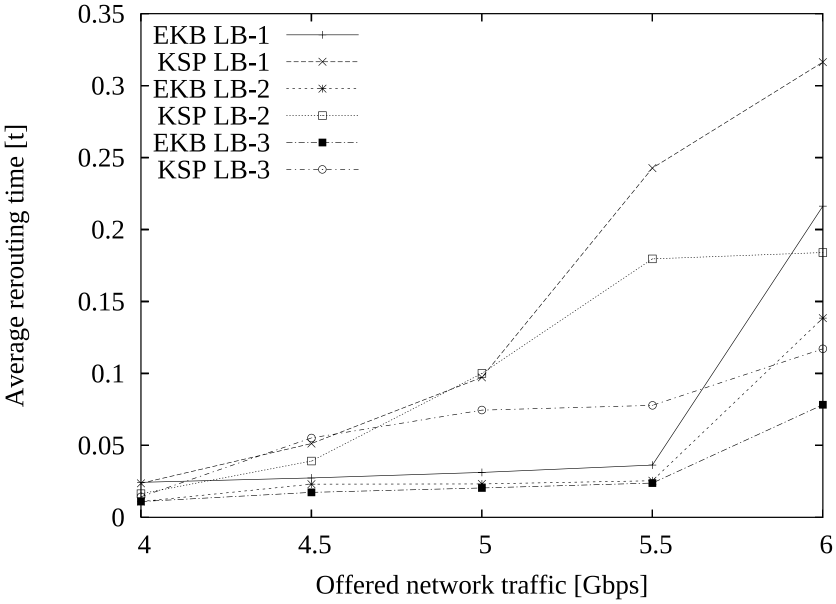

Figure 7: Average rerouting time values for $k = 3$.

Thesis 3: Cost-efficient core network design [C5, J1, J4, B1]

In the recent years a wide range of telecommunication services became available for customers aiming at facilitating their everyday activities. Not only does the penetration of existing applications increase, the demand for new types of services arises as well. Thus, the need for network capacity increases continuously. Although the extension of existing networks can help in some cases, both the significant increase of traffic and the deployment of novel services require the establishment of new networks in general. Further, another situation when a new network is needed may be the appearance of a novel operator, or a new type of technology.

In the current interpretation the establishment of a network means that network layer communication is provided between various locations corresponding to a given organization. To realize such a service two main components are essential: (1) links that establish physical connectivity between the locations, and (2) routers that provide network layer services in the locations. The establishment cost of a network largely depends on the goodness of the design phase. The two main characteristics of an optimally designed network is that routers have the appropriate capacities to fulfill the traffic demands, and links are properly dimensioned. The relationship between cost and capacity values can be represented with the help of monotonic nondecreasing *cost functions.*

Several approaches regarding the applied cost model co-exist in network design, which differ in mathematical complexity and accuracy. The linear function based approximations are inadequate to model real cost relations in the overwhelming majority of cases. However, it is a relatively frequent approach because of the easier computations involved [PMJ+01a, PMJ+01b]. Practically, the real devices have standard capacity values as they are built up from smaller modules with fixed capacity units. These types of cost dependences can be described with so-called *stepwise* functions [HJS99, CHG00, CS00, PF02].

The use of stepwise cost functions represents a sophisticated method that enables the modeling of real cost/capacity relations with adequate accuracy. However, this non-linear approach implies the high computational complexity of the design task, i.e., the problem itself is NP-hard [GJ79]. Moreover, the formulation of the problem is also a very complex task. Therefore, heuristic approaches are preferred in this area aiming at providing a favorable solution while having acceptable running time.

Cost-efficient network design is a problem that arises from time to time, thus, numerous valuable research activities focus on this area [Min89, PMJ+01a, PMJ+01b, CHG00]. However, they are similar in the sense that they do not emphasize the importance of stepwise cost functions. The feature of the novel network design algorithm proposed here is that it applies heuristics that can benefit from the specialties of stepwise cost functions during the optimization.

Thesis 3.1: Core network design algorithm

I have proposed a new algorithm that can solve the cost-efficient network design task efficiently when applying the stepwise cost model. The new core network design algorithm (CND) combines the benefits of global and local search algorithms in order to get closer to the optimal solution, while aiming at taking advantage of the characteristics of stepwise cost functions.

The algorithm can be divided into three subsequential phases. The process starts with the initial capacity estimation (ICE) phase, whose task is to estimate the approximate capacity conditions. The main phase is the iterative routing optimization (IRO), which searches for an appropriate network configuration based on the initial estimations. Although this second phase already provides an economical solution to the network design problem, with the help of the last phase, namely, the posterior capacity refinement (PCR) method a cheaper configuration can be reached.

As mentioned above, IRO is the main phase, which can provide a valid solution for the problem, while the use of ICE and PCR is optional. However, in spite of that they increase the running time it is worth using them as they contribute significantly to the quality of the solution.

Initial capacity estimation. The main purpose of the initial capacity estimation (ICE) phase is to foresee the necessary device capacities by analyzing the set of traffic demands. Thus, prior information can be provided for IRO by supplying a partially dimensioned network as an initial state. Therefore, the remaining task is to finalize the dimensioning of devices and the routing of demands.

The operation of ICE consists of four steps. In step 1 the traffic demands to be accommodated are shuffled randomly. Then, in step 2 they are routed one after another (in the above order) using such a weight function that prefers devices with low cost per unit traffic aiming at keeping the overall network cost at a low level. These steps are repeated a predefined number of times (step 3). Finally, in step 4 the capacities of devices are specified in such a way that they get the lowest value of the ones arising in the above accommodations. This idea is based on the assumption that if all the random accommodations needed a certain amount of resources on a particular device, then it is probable that the optimal accommodation also needs so much resources.

Iterative routing optimization. The main phase of the algorithm, namely, the iterative routing optimization (IRO) phase is based on an algorithm that can accommodate a given set of traffic demands in a capacitated graph. In the current investigation the algorithm proposed in [JKM+01] is used for this purpose. An important advantage of IRO is that the routing optimization procedure applied can be substituted by any other method performing the same task.

The main steps of IRO are the following. In step 1 IRO attempts to accommodate the traffic demand set considering the actual capacity constraints. If the algorithm terminates with success (step 2), i.e., all demands are accommodated, IRO finishes. Otherwise, the capacity of a particular device is increased by one capacity step in the following way. In step 3 the remaining demands that could not be fit into the graph under the actual capacity conditions are accommodated disregarding the capacity constraints. The device to be extended in step 4 is the one on which the capacity violation, i.e., the extra capacity required by the actual accommodation is the largest. After the capacity increase, step 1 follows again with the updated capacity limits.

Posterior capacity refinement. As discussed earlier, after the IRO phase a suitable solution to the network design problem is already available. However, this result might be improved with the help of posterior capacity refinement (PCR). This greedy method is based on a local search procedure, which means that the process concentrates only on one part of the network at a time. The idea behind PCR is to reduce the size of devices that are underutilized in the sense that if a relatively small amount of traffic was removed from them, a device with lower capacity and consequently lower cost level would suffice.

The operation of the PCR method has six steps. In step 1 the devices are sorted by their relative step utilization, i.e., the utilization of the capacity range belonging to the current cost value. The reason for this is that in the case of a device with lower relative step utilization it is more probable that if its capacity is decreased by one step, the traffic demands can still be satisfied under these tighter capacity conditions. Then, in step 2 the devices are shrunk one by one in the above order, and the traffic demands are tried to be accommodated by IRO in step 3. If the shrinking of a particular device results in such a configuration that is cheaper than the best overall cost so far (step 4), the process restarts from the sorting step. Otherwise, the solution before the capacity decrease step is restored in step 5. If there are more devices that were not shrunk after the last overall cost improvement the process continues from step 2, else the process terminates (step 6).

Thesis 3.2: Performance evaluation of the core network design algorithm

I have conducted simulations to investigate the efficiency of the proposed algorithm CND. The experiments were focused on the measures total network cost and running time since they can be considered main performance indicators in the case of heuristic network design algorithms.

Since the network design task deals with future networks in general, random topologies were used in the problem instances [JO99, JKM+01]. Various types of topologies were examined during the simulations, however, results presented here focus on 15, 25, and 35-node networks with an average initial nodal degree of 5. For a given topology, various traffic demand sets were tested, i.e., the distribution as well as the volume of the traffic were varied.

In the case of links the standard capacity values of STM were taken as a starting-point, i.e., 155 Mbps, 622 Mbps, 2.5 Gbps, and 10 Gbps. Further, supposing that the price of two pieces of a particular device is less than the price of the next device, the establishment of two same devices parallel was also allowed. Thus, eight possible capacity values could be chosen from per link. Assuming that the number of different size of routers is limited, three capacity levels were distinguished specifying the maximum capacity to be managed by the given router as 1, 10, or 100 Gbps.

In order to evaluate the new algorithm CND, a comparison to a well-known, published method was performed. For this purpose a greedy random adaptive search procedure (GRASP) [RR02] based algorithm was used. In [KSW+01] it was shown that with the help of the GRASP paradigm the location and sizing problems in network design can be solved efficiently.

Total network cost. Table 4 shows the average cost ratios for different network sizes. The proposed full algorithm (ICE+IRO+PCR) provided the most economical network plans in the current simulation scenarios. IRO—that is capable of solving the design task by itself—achieved about 12-20% worse results. By applying ICE, this value is reduced to 9-13%, while the IRO+PCR combination achieved less than 4% higher average network costs than the best, i.e., the full algorithm. Comparing the proposed algorithm with the reference algorithm GRASP it can be concluded that in the case of 25-node and 35-node networks the proposed algorithm performed similarly providing about 3% cost reduction, while in the case of 15-node networks the difference was a bit higher, more than 5%.

Network size	Algorithm				
[node]	IRO	ICE+IRO	IRO+PCR	ICE+IRO+PCR	GRASP
15	111.57	108.52	101.68	100.00	105.57
25	116.10	109.62	103.77	100.00	103.27
35	120.41	112.65	102.28	100.00	103.49

Table 4: Ratio of total network costs for various network sizes [%].

Running time. An important issue during the characterization of a novel optimization algorithm is the running time. Table 5 depicts the values measured on a Sun Ultra Enterprise 420R machine with an Ultra II 450 MHz processor and 1 GByte of RAM. The first fact that can be observed is that the time consumption of the reference algorithm GRASP was significantly higher than that of the novel algorithms. Another observation is that the running time increases about tenfold when increasing the network size from 15 nodes to 25 nodes, while jumping to 35 nodes the raise is more moderate.

Network size [node]	Algorithm				
	IRO	ICE+IRO	IRO+PCR	ICE+IRO+PCR	GRASP
15	14	20	50	54	5336
25	102	241	471	579	50018
35	282	1088	2591	4009	188941

Table 5: Running time values [s].

Thesis 4: Cost-efficient VoIP network design [C9, J4, B1, C11]

Nowadays, the all-IP concept is favored by the infocommunication industry, intending to conduct all different types of traffic over the Internet protocol dominant in the networking area. As part of all-IP an increasing number of companies in the telephony area commit themselves to using the voice over IP (VoIP) technology. Assuming a large VoIP network having a huge amount of customers it is necessary to take QoS as well as economic goals into account during the design phase.

In the model applied the VoIP network consists of two logical components: the access network, and the transport core network. The access network includes the *VoIP nodes*, namely, the customer end-points intending to use the VoIP service. The transport network serves for carrying the aggregated VoIP traffic between the various access areas. Its main parts are the edge core routers, the transit core routers, and the physical links between them.

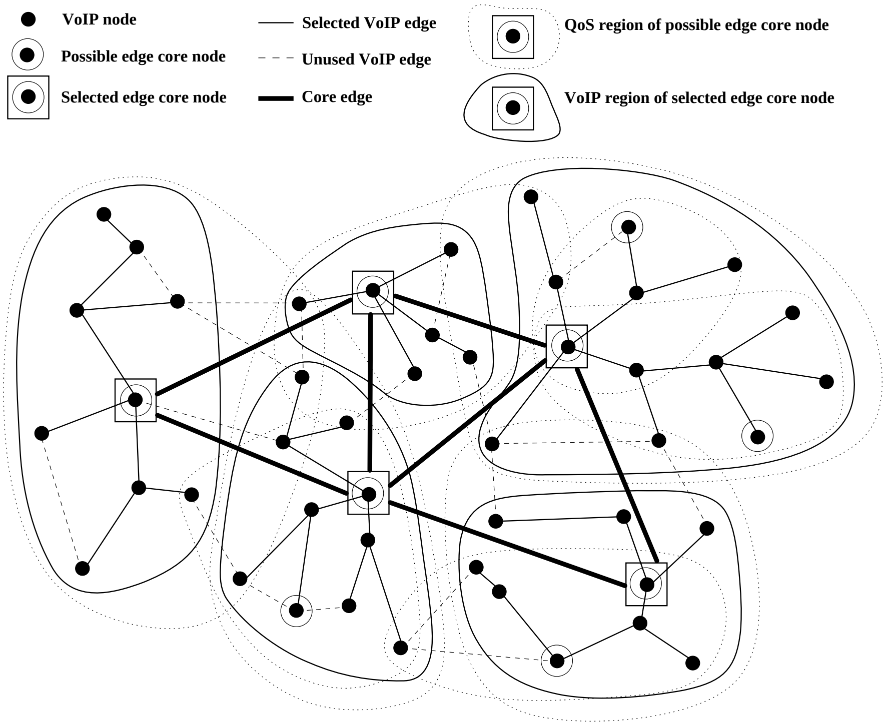

Figure 8: QoS and VoIP regions of a VoIP network.

A VoIP node can reach more than one edge core router fulfilling the QoS requirements (e.g., limited maximal delay) of the telephone service. Therefore, it is necessary to select one of them that will serve as a gateway towards the transport network. The VoIP nodes assigned to the same edge core router form a so-called *VoIP region*. After specifying the VoIP regions, the gateways selected have to be connected resulting in a transport core network. The cost of this network depends on the capacity of the corresponding devices, such as routers and links.

In summary, two main tasks can be differentiated in the case of VoIP network design: (1) VoIP region specification, i.e., the assignment of each VoIP node to exactly one core edge router, and (2) the design of the transport core network covering the selected gateway nodes. These two tasks can be solved independently by applying existing methods, which means that the objective of transport core network design is disregarded during VoIP region specification. However, it can be more efficient to take cost and quality factors concerning the transport network to be composed into consideration already in the first task. In this thesis the first task is addressed by making several propositions to solve the VoIP region specification task. In order to solve the transport core network design problem, the core network design algorithm (CND) presented in Thesis 3 is used after specifying the VoIP regions.

Thesis 4.1: VoIP region specification algorithm

I have proposed a novel algorithm that can efficiently solve the VoIP region design subproblem. The main feature of the concept is that the objective function of the transport network design is taken into consideration during the VoIP region specification phase.

The two main components of the algorithm are: (1) a combinatorial optimization metaheuristic that scans the state space of possible solutions, and (2) a metric that describes the goodness of various solutions. Two different optimization metaheuristics are examined, namely, the evolutionary algorithm and simulated annealing paradigms. The quality of the solution of either metaheuristic is heavily influenced by the cost calculation method used for evaluating the goodness of the current solution, thus, more metrics are investigated.

Evolutionary covering algorithm (EC). One approach considered here is a covering algorithm that is based on the well-known paradigm of *evolutionary algorithms* (EC) (also called *genetic algorithms*) [BFM96], which enables selection between more feasible solutions using complex *cost calculation methods*. The representation of the VoIP region specification subproblem applied in the evolutionary algorithm is the following. An *entity* defines a valid assignment, where each *gene* corresponds to a VoIP node, and its value refers to an available core node the particular VoIP node is assigned to.

During *mutation* one gene of an entity is changed randomly, which means that the given VoIP node is reassigned to another available core node. During the *crossover* of two entities, the child inherits its genes from either of its parents with equal probability. The selection of entities that do not *survive* is performed by killing the oldest entity from m randomly selected entities, and killing the worst one (based on the cost calculation method used) from another m randomly selected entities.

In order to create an appropriate *initial population*, the GC-C algorithm is applied, where GC-C refers to the core node cost based version of the greedy covering algoritm presented in [D95, J4]. Several feasible solutions are sought by GC-C excluding a different core node from the initial set of core nodes in each iteration. Naturally, core nodes that are mandatory elements of the solution cannot be excluded. Using this method several different feasible coverings consisting of core nodes having low establishment cost values can be generated to form the initial population. The evolutionary algorithm stops if the cheapest solution considering the actual cost calculation has not changed during the last i steps.

Simulated annealing based covering algorithm (SC). Another possible solution to the VoIP region specification subproblem is a covering algorithm that is based on the principle of *simulated annealing* (SC) [JAM+91, KGV83]. The representation used is similar to that presented above, namely, a *state* defines a valid assignment between the VoIP nodes and the core nodes, consisting of values each of which refers to the core node the particular VoIP node is assigned to. The *initial state* is based on a solution by GC-C, while *random changes* in the state are generated by reassigning a randomly selected VoIP node to another available core node.

States are evaluated using the same cost calculation methods as in the case of the evolutionary algorithm. A new state is accepted with a probability of $e^{\frac{C-C'}{T}}$, where C and C' denote the cost of the previous and new state, respectively, while T is the *temperature* of the previous state. The *annealing schedule* takes the form of $T' = T \cdot \phi$, where T is the temperature in the previous iteration, T' is the new temperature, and ϕ denotes the *annealing factor*. The *initial temperature* is denoted by T_0. As seen in the case of the evolutionary algorithm, the simulated annealing algorithm terminates if the cost of the cheapest solution has not improved during the last i iterations.

Entity and state cost calculation methods. The quality of the solution of both EC and SC is heavily influenced by the cost calculation method used for evaluating the entities and states, respectively. Thus, more approaches are investigated, as it can be seen in the following paragraphs.

The main idea behind the *cost approximation* based methods (EC-C and SC-C) is to try to foresee the final cost of the transport core network to be designed. The set of the aggregated VoIP traffic demands are routed several times, based on different random orders. *Dijkstra's shortest path algorithm* is applied for this purpose with an edge weight function that prefers devices having low cost per unit traffic values. Finally, the price of the cheapest configuration is considered the cost of the entity or state.

When applying the *distance weighted traffic* based methods (EC-D or SC-D), the product of the bandwidth requirement and the length of the possible shortest path (in terms of hop-count) is calculated for each aggregated VoIP traffic demand. Then, the cost of the entity or state is specified as the sum of these products regarding the whole network.

The two-level variants of the above cost metrics (EC-C2, EC-D2, SC-C2, SC-D2) were also investigated in the following way. The number of selected core edge routers (gateways) serves as primary metric. After that, the value computed by the cost approximation or distance weighted traffic methods normalized by the initial value of the metric and scaled by an importance factor s is used as a secondary metric.

The *interconnection* based methods (EC-I and SC-I) aim at reducing the total network cost by selecting a group of heavily interconnected edge core nodes, thus, diminishing the number of transit core nodes required. The degree of interconnection is defined as the number of possible core edges connecting the selected core nodes directly divided by the theoritical maximal number of edges referring to the case when the selected core nodes are fully meshed. The cost of an entity or state is then defined as the number of selected core nodes divided by the degree of interconnection.

In the case of the *traffic weighted interconnection* based methods (EC-TI and SC-TI), each core edge between the currently selected core nodes is assigned a weight defined as the sum traffic of the two core nodes it connects. The weights of all possible core edges between the selected core nodes are summed, and this sum is divided by its theoritical maximal value referring to a fully interconnected group of core nodes. The cost of the entity or state is defined as the number of selected core nodes divided by the above ratio. The idea behind this metric is to prefer assignments where core nodes with higher traffic values are more heavily interconnected, while taking the number of selected core nodes into consideration at the same time.

Thesis 4.2: Performance evaluation of the VoIP region specification algorithm

I have carried out simulations to analyze the behavior of the various algorithm proposals. During the simulations the total network cost value was considered the principal performance measure. However, the number of core nodes as well as running time consumption factors were also examined.

The first task of problem instance creation is to generate the topology of the network including the VoIP nodes and edges as well as the possible core nodes and edges. For this purpose a random VoIP graph generator method was applied that is based on the *Barabási-Albert model* [BA99, AB02]. This approach is based on the *power laws* of Internet topology [FFF99, BT02], and nowadays it is frequently used to model wide area communication networks. Topologies of various sizes were examined, however, results are presented for networks having 500 VoIP nodes, and 50 possible core nodes.

The use of the codec G.711 PCM—recommended by the International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) [ITU]—with silence suppression having an activity factor of 55% and a packetization time of 5 ms was assumed. In this way, the investigated average VoIP node traffic interval was 2 to 16 Mbps. Obviously, the proposed design approach can also handle any other codec types such as G.723.1, G.726, or G.729A.

During the transport core network design the cost model presented in Thesis 3 was followed. In this way, the cost functions of core edges were based on the STM standards, while in the case of core nodes three different size of devices were assumed. The detailed description of the used cost functions can be found in [J4].

During the simulations the GC-N and GC-C construction algorithms were applied as reference, where GC-N and GC-C refer to the core node number based and the core node cost based versions of the greedy covering algorithm presented in [D95, J4].

Total network cost. Figure 9 shows the total network cost for the two variants of the greedy covering algorithm GC used as references, and the two best variants of both EC and SC. As it can be seen in the figure, GC-C provided 3-5% lower network costs than the simplest algorithm GC-N. The more sophisticated optimization algorithms EC and SC proved that more efficient results can be achieved by selecting from multiple feasible solutions, as they outperformed the greedy covering algorithms in almost all cases. While the two variants of the evolutionary algorithm EC achieved better results than the methods based on simulated annealing at lower average VoIP node traffic values, above 12 Mbps the results of EC and SC converged. However, EC-D2 proved to be the best algorithm overall providing an improvement of 7-20% compared to the greedy algorithm GC-N over the traffic interval investigated.

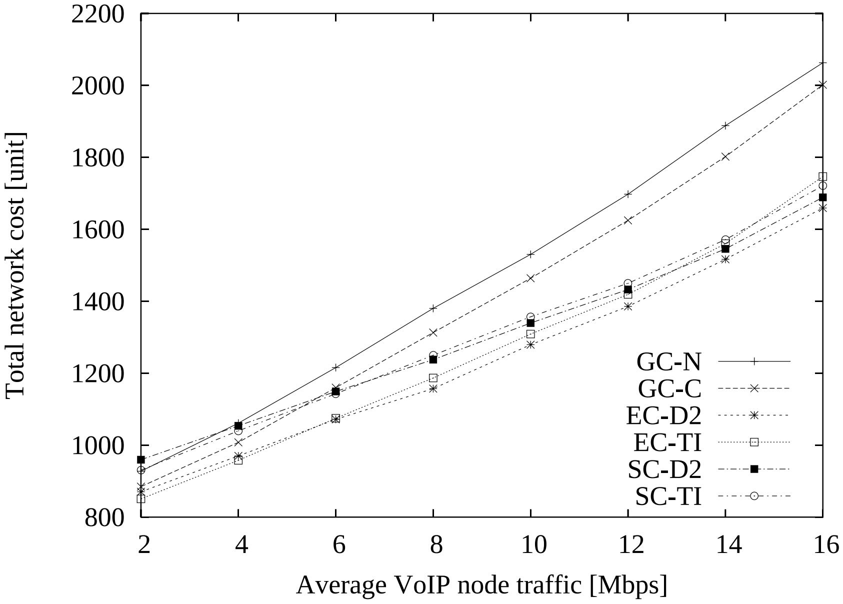

Figure 9: Total network cost using different algorithms.

Considering the total network cost values it can be observed that the dispersion of the results of various algorithms is larger in the higher traffic range. The reason for this is that more different devices varying in capacity have to be chosen from in this case, which results in the increase of the size of state space.

Number of core nodes. Besides the cost of the transport core network its size is also an important attribute, which can be described well by the number of core nodes. Thus, in this scenario the algorithms were compared focusing on this basic measure. An important observation in Figure 10 is that in the case of both EC and SC the number of edge core nodes is higher for the two-level distance weighted traffic based variant than the traffic weighted interconnection based method, while the relation between the total network cost values of the two variants is exactly the opposite. This means that the sophisticated selection of VoIP traffic aggregation points is more important than keeping their number as low as possible.

Running time. Although in the case of off-line network design the running time has only secondary importance, it is worth examining this factor also in order to make the investigations complete. Figure 11 depicts the values for the two versions of the greedy covering algorithm GC and the best two variants of both EC and SC measured on a Sun Ultra Enterprise 420R machine with an Ultra II 450 MHz processor and 1 GByte of RAM. As it can be seen, the GC algorithms were very fast as they provided results within one minute on average. Another important point to note is that for both cost calculation methods shown the running time value of the simulated annealing based algorithm SC was higher than that of the evolutionary algorithm EC.

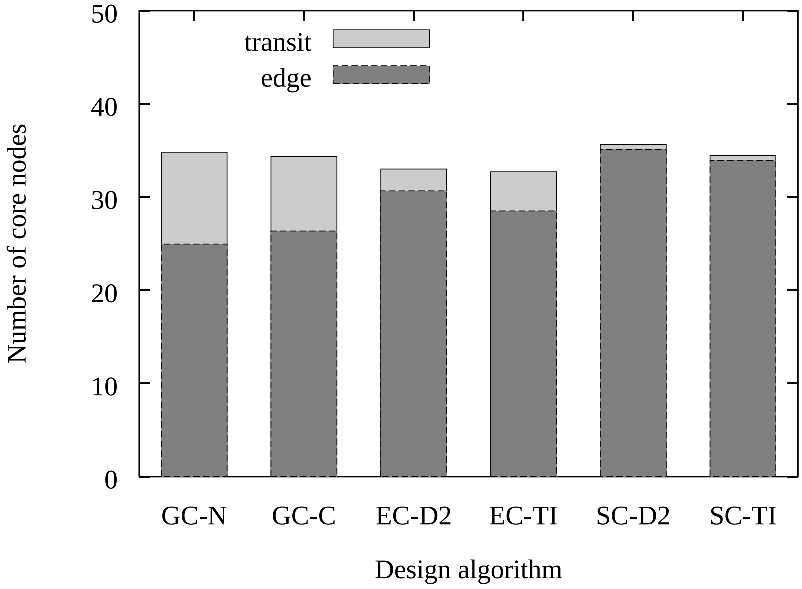

Figure 10: Number of edge and transit core nodes using different algorithms.

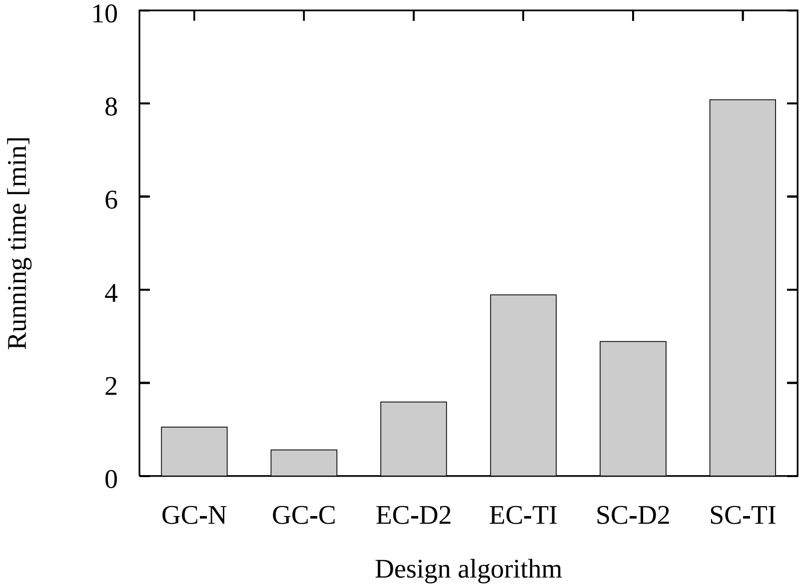

Figure 11: Running time values of the different algorithms.

3 Application of the results

Besides the scientific significance, the research work presented in the dissertation also has a relevant industrial motivation considering that all new traffic engineering algorithms proposed were initiated by projects developing new products and/or performing applied research activities. Prompt partial path optimization was originally intended to be a function in an IP management tool developed by Ericsson Hungary Ltd. The improvement of the precalculated paths based routing concept was related to activities examining the routing capabilities of an Ericsson switch. The core network design algorithm was ordered by an Ericsson project developing a transport network optimization tool. Finally, the VoIP network planning task was included in the project IKTA-0092/2002. of the Ministry of Education, Hungary—having participants from Ericsson Hungary Ltd., Budapest University of Technology and Economics, and Kovax 95 Ltd.—that addressed the design of a complex VoIP network management system.

References

[AAF+93] 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 the 25th Annual ACM Symposium on Theory of Computing, pp. 623–631, 1993.

[AB02] R. Albert and B. Barabási, *Statistical Mechanics of Complex Networks*, Review of Modern Physics, vol. 74, pp. 47–97, January 2002.

[AKK+00] P. Aukia, M. Kodialam, P. V. Koppol, T. V. Lakshman, H. Sarin, and B. Suter, *RATES: a server for MPLS traffic engineering*, IEEE Network, vol.14, no. 2, pp. 34–41, March-April 2000.

[Atm02] ATM Forum Technical Commitee, *Private Network–Network Interface Specification Version 1.1*, ATM Forum af-pnni-0055.002, April 2002.

[AWK+99] G. Apostolopoulos, R. Williams, S. Kamat, R. Guerin, A. Orda, and A. Przygienda, *QoS Routing Mechanisms and OSPF Extensions*, Internet Engineering Task Force (IETF), Request For Comments (Proposed Standard) 2676, August 1999.

[BA99] A. Barabási and R. Albert, *Emergence of scaling in random networks*, Science, pp. 509–512, October 1999.

[BD97] M. Berkelaar and J. Dirks, `lp_solve 2.2`, `ftp://ftp.es.ele.tue.nl/pub/lp_solve`.

[BFM96] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*, University Oxford Press, New York, 1996.

[BT02] T. Bu and D. Towsley, *On Distinguishing between Internet Power-Law Topology Generators*, In proc. of INFOCOM'2002, June 2002.

[CHG00] T. Cinkler, T. Henk, and G. Gordos, *Stochastic Algorithms for Design of Thrifty Single-Failure-Protected Networks*, Proc. of Design of Reliable Communications Networks, DRCN'2000, Munich, Germany, 2000.

[CS00] S. Chamberland and B. Sansò, *Overall Design of Reliable IP Networks with Performance Guarantees*, Les Cahiers du GERAD, vol. 30-2000.

[D95] M. S. Daskin, *Network and Discrete Location*, Wiley, New York, 1995.

[EK72] J. Edmonds and R. M. Karp, *Theoretical improvements in algorithmic efficiency for network flow problems*, Journal of ACM 19, pp. 248–264, 1972.

[FF56] L. R. Jr. Ford and D. R. Fulkerson, *Maximal Flow Through a Network*, Canadian Journal of Mathematics, pp. 399–404, 1956.

[FFF99] M. Faloutsos, P. Faloutsos, and C. Faloutsos, *On Power-Law Relationships of the Internet Topology*, Proc. of ACM SIGCOMM'99, August 1999.

[FJM+00] J. Forslöw, I. Jarrett, P. Moran, and B. Szviatovszki, *Management solutions for IP networks*, Ericsson Review, vol. 77, no. 1, pp. 42–52, 2000.

[GJ79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.

[HJS99] J. Harmatos, A. Jüttner, and Á. Szentesi, *Cost-based UMTS Transport Network Topology Optimisation*, Proc. of International Conference on Computer Communications (ICCC'99), Tokyo, Japan, pp. 00111-1-8, 1999.

[ICM+02] G. Iannaccone, C-N. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, *Analysis of link failures over an IP backbone*, Proc. of ACM SIGCOMM Internet Measurement Workshop, Marseilles, France, November 2002.

[ITU] International Telecommunication Union – Telecommunication Standardization Sector, `http://www.itu.org/ITU-T`.

[JAM+91] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, *Optimization by Simulated Annealing: an Experimantal Evaluation*, Operations Research, vol. 39, no. 1, May-June 1991.

[JKM+01] B. G. Józsa, Z. Király, G. Magyar, and Á. Szentesi, *An Efficient Algorithm for Global Path Optimization in MPLS Networks*, Optimization and Engineering, vol. 2, no. 3, pp. 321–347, 2001.

[JM03] B. G. Józsa and M. Makai, *On the Solution of Reroute Sequence Planning Problem in MPLS Networks*, Computer Networks, vol. 42, no. 2, pp. 199–210, June 2003.

[JO99] B. G. Józsa and D. Orincsay, *Random Graph Generator (for telecommunication networks)*, Technical Report, Ericsson Research, TrafficLab, Budapest, Hungary, 1999.

[KA00] K. Kompella and D. Awduche, *Notes on Path Computation in Constraint-Based Routing*, Internet Engineering Task Force (IETF), Internet Draft, September 2000.

[KL00] M. Kodialam and T. V. Lakshman, *Minimum Interference Routing with Applications to MPLS Traffic Engineering*, Proc. of IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000.

[KGV83] S. Kirkpatrick, C. D. Gerlatt Jr., and M. P. Vecchi, *Optimization by Simulated Annealing*, Science, no. 220, pp. 671–680, 1983.

[KSW+01] K. Kumaran, A. Srinivasan, Q. Wang, S. Lanning, and K. G. Ramakrishnan, *Efficient Algorithms for Location and Sizing Problems in Network Design*, Proc. of Global Communications Conference (GLOBECOM'2001), 2001.

[Lil03] C. Liljenstolpe, *Offline Traffic Engineering in a Large ISP Setting*, Internet Engineering Task Force (IETF), Internet Draft, March 2003.

[Min89] M. Minoux, *Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications*, Proc. of Networks 19, pp. 313–360, 1989.

[MSS00] Á. Magi, Á. Szentesi, and B. Szviatovszki, *Analysis of Link Cost Functions for PNNI Routing*, Computer Networks, vol. 34, no. 1, pp. 181–197, July 2000.

[MSZ96] Q. Ma, P. Steenkiste, and H. Zhang, *Routing High-Bandwidth Traffic in Max-Min Fair Share Networks*, Proc. of ACM SIGCOMM, pp. 206–217, August 1996.

[NSB+02] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot, *IS–IS Link Weight Assignment for Transient Link Failures*, Sprint ATL Technical Report Nr. TR02-ATL-071000, Sprint Labs, July 2002.

[PF02] M. Prytz and A. Forsgren, *Dimensioning of a Multicast Network that Uses Shortest Path Routing Distribution Trees*, Technical Report, TRITA-MAT-02-OS01, Department of Mathematics, Royal Institute of Technology, February 2002.

[PG94] M. Pióro and P. Gajowniczek, *Simulated Allocation: a Suboptimal Solution to the Multicommodity Flow Problems*, Proc. of the 11th UK Teletraffic Symposium, 1994.

[Plo95] S. Plotkin, *Competitive Routing of Virtual Circuits in ATM Networks*, IEEE Journal of Selected Areas in Communications, vol. 13, no. 6, pp. 1128–1136, August 1995.

[PMJ+01a] M. Pióro, A. Mysłek, A. Jüttner, J. Harmatos, and Á. Szentesi, *Topological Design of MPLS Networks*, Proc. of Global Communications Conference (GLOBECOM'2001), San Antonio, Texas, USA, 2001.

[PMJ+01b] M. Pióro, A. Mysłek, A. Jüttner, J. Harmatos, Á. Szentesi, and P. Gajowniczek, *Topological Design of Telecommunication Networks – Nodes and Links Localization under Demand Constraints*, Proc. of the 17th International Teletraffic Congress, Salvador de Bahia, 2001.

[RR02] M. G. C. Resende and C. C. Ribeiro, *Greedy Randomized Adaptive Search Procedures*, to appear in State of the Art Handbook in Metaheuristics, 2002.

[SRS01] A. Shaikh, J. Rexford, and K. G. Shin, *Evaluating the Overheads of Source-Directed Quality-of-Service Routing*, IEEE/ACM Transactions on Networking, vol. 9, issue 2, pp. 162–176, April 2001.

[SSJ01] B. Szviatovszki, Á. Szentesi, and A. Jüttner, *Minimizing Re-routing in MPLS networks with preemption-aware constraint-based routing*, Proc. of the 2001 International Symposium of Performance Evaluation of Computer and Telecommunication Systems, pp. 249–261, July 2001.

[ST84] J. W. Suurballe and R. E. Tarjan, *A Quick method for Finding Shortest Pairs of Disjoint Paths*, Networks, vol. 14, pp. 325–336, 1984.

[Top98] D. M. Topkis, *A k Shortest Path Algorithm for Adaptive Routing in Communication Networks*, IEEE Transactions on Communications, vol. 36, no. 7, July 1998.

[Wan98] D. Wang, *Network Planning and Analysis Tools for MPLS Networks*, Slide presentation given at the First International Workshop on Multiprotocol Label Switching (IW-MPLS'98), November 1998.

[WC96] Z. Wang and J. Crowcroft, *Quality-of-Service Routing for Supporting Multimedia Applications*, IEEE Journal on Selected Areas in Communications (JSAC), 14(7) pp. 1288–1234, September 1996.

[Yen71] J. Y. Yen, *Finding the K Shortest Loopless Paths in a Network*, Management Science 17, pp. 712–716, 1971.

Publications

Journal papers

[J1] D. Orincsay and B. G. Józsa, *Távközlési hálózatok költséghatékony tervezése*, Híradástechnika, vol. 58, no. 4, pp. 39–45, April 2003.

[J2] D. Orincsay, B. Szviatovszki, and G. Böhm, *Prompt Partial Path Optimization in MPLS Networks*, Computer Networks (Elsevier), vol. 43, issue 5, pp. 557–572, December 2003.

[J3] L. Tamási, B. G. Józsa, and D. Orincsay, *Távközlési hálózatok tervezése a forgalomeloszlás változásainak figyelembevételével*, Híradástechnika, vol. 59, no. 9, pp. 32–39, September 2004.

[J4] L. Tamási, D. Orincsay, and B. G. Józsa, *Cost-Optimal Design of VoIP Networks Using the VPN Concept*, Computer Networks (Elsevier), 2005.

Conference papers

[C1] B. G. Józsa, D. Orincsay, Á. Magi, and B. Szviatovszki, *On the Use of Trunk Reservation in PNNI Routing*, Proc. of the IEEE Conference on High Performance Switching and Routing Technologies (3rd International Conference on ATM – ICATM'2000), Heidelberg, Germany, pp. 135–139, June 2000.

[C2] A. Jüttner, B. Szviatovszki, Á. Szentesi, D. Orincsay, and J. Harmatos, *On-demand optimization of label switched paths in MPLS networks*, Proc. of the 9th International Conference on Computer Communication and Networks (IC-CCN'2000), Las Vegas, USA, pp. 107–113, October 2000.

[C3] D. Orincsay and J. Harmatos, *On-demand optimization of protected LSP-tunnels in MPLS networks*, Proc. of the 9th IFIP Working Conference on Performance Modelling and Evaluation of ATM & IP Networks (IFIP ATM & IP 2001), Budapest, Hungary, pp. 357–368, June 2001.

[C4] B. G. Józsa and D. Orincsay, *Shared Backup Path Optimization in Telecommunication Networks*, Proc. of the 3rd IEEE International Workshop on Design of Reliable Communication Networks (DRCN'2001), Budapest, Hungary, pp. 251–257, October 2001.

[C5] B. G. Józsa, D. Orincsay, and 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 2003.

[C6] D. Orincsay, B. G. Józsa, D. Kósa, and L. Tamási, *Recovery Using Precalculated Paths Based Routing in ATM Networks*, Proc. of the 10th Computer Networks Conference, Szczyrk, Poland, pp. 119–128, June 2003.

[C7] B. G. Józsa, D. Orincsay, and 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 2003.

[C8] B. G. Józsa, D. Orincsay, and 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 2004.

[C9] D. Orincsay, B. G. Józsa, and 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 2004.

[C10] D. Orincsay, B. G. Józsa, and L. Tamási, *On the Use of EKB Routing Strategy in PNNI Networks*, Proc. of the IEEE International Conference on Communications (ICC'2004), vol. 2, pp. 1003–1007, Paris, France, June 2004.

[C11] L. Tamási, D. Orincsay, B. G. Józsa, and G. Magyar, *Design of Survivable VPN Based VoIP Networks*, submitted, 5th IEEE International Workshop on Design of Reliable Communication Networks (DRCN'2005), Island of Ischia, Italy, October 2005.

Book chapters

[B1] D. Orincsay, L. Tamási, B. G. Józsa, and G. Magyar, *Design of Survivable VoIP Networks Using the VPN Concept*, preliminarily accepted, Nova Science Publishers, January 2005.