OPTIMIZATION METHODS FOR VIRTUAL PRIVATE NETWORK DESIGN

Collection of Ph.D. Theses

By
Markosz Maliosz

Research Supervisor:

Tibor Cinkler Ph.D.
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
BUDAPEST, HUNGARY
2005

1 Introduction

Virtual networks are basically logical overlays on a physical network. The broadest definition of a VPN is "any network built upon a public network and partitioned for use by individual customers" [SOL].

There are various types of VPNs according to the International Standards Organization Open System Interconnection (ISO - OSI) reference model [fS94].

- Layer 1 (physical layer) VPN is a virtual private network made up of leased circuits connecting customer sites. Examples of layer 1 connections are optical (OVPN) or time division multiplexing (TDM) paths.
- Layer 2 (data link layer) VPN supplies a layer 2 point-to-point service or emulates LAN service across a Wide Area Network (WAN) [AR04]. Examples for layer 2 VPNs are frame relay, X.25, and ATM networks.
- Layer 3 (network layer) VPNs are implemented with point-to-point IP-over-IP tunnels constructed across shared IP backbones, referred to as "IP VPNs" [GLH+00, CS05]. Examples for tunneling are the IPSec [KA98] or GRE [FLH+00] protocol frameworks.

Thus, the term VPN is used in different networks (ATM, Frame Relay, MPLS, etc.) and by tunneling protocols too (GRE, IPsec, PPTP [HPV+99], L2TP [TVR+99], etc.).

The term "private" has at least two interpretations. For ATM and Frame Relay, private means that the virtual circuits connect a closed group or community of users. For IPsec and other protocols it means that they use cryptography to provide authentication and confidentiality, thus the term "private" still means a closed user group. But it also means message confidentiality: traffic is encrypted so that no "man in the middle" can capture or modify information passed site-to-site or user-to-user.

The goal of all VPN products is to enable deployment of logical networks, independent of physical topology, allowing a geographically distributed group of hosts to interact and be managed as a single network.

One type of VPN application is *user-to-user* remote access for joint project workers, or for a home user to log on to the company intranet. The other typical application is *site-to-site* VPN that connects fixed sites to a corporate LAN, thus extending it over a public or shared network.

A VPN incorporates two features, encryption and tunneling, to ensure that the data is delivered safely and privately across the public space. The traffic is encrypted at the edge of one network or at the originating computer, moved over the public shared network like any other data, and then decrypted when it reached the corporate network or a receiving computer. This encrypted traffic acts like it is in a tunnel between the two networks. The use of encryption assures the *security* for VPNs.

Quality of Service (QoS) is a requirement for any VPN deployment where performance is important. QoS enabled VPNs are able to emulate a private wide area network using IP facilities and guarantee bandwidth and latency. However, the complexities introduced by VPNs and the requirement to provide QoS can make the job of the ISPs and systems administrators extremely difficult.

The VPN solution over public network infrastructure gives a cheaper and more flexible alternative compared to leased lines, therefore VPN services became more popular recently.

2 Objectives of the Research

VPNs using the public Internet today are limited to "best effort". However, large providers can offer high-performance VPNs. These premium VPN services increase cost, but guarantee QoS by routing the traffic over the ISP's own access links and backbone network. As ISPs roll out quality of service using e.g. MPLS (multiprotocol label switching), they will begin offering "differentiated" VPN services that can span multiple networks. Customers will be able to choose – and pay for – either best effort or guaranteed service levels (e.g. low latency, low loss, committed throughput). The thesis work focuses on VPNs provided by ISPs, that can apply design techniques to offer bandwidth guaranteed VPNs, and not on Internet VPNs.

VPNs need to provide private, ubiquitous communications to the locations and users that require it. It must do this in a secure manner while maintaining as many of the characteristics of traditional private WAN connections as possible. This thesis work focuses not on the security issues, but the topological design of VPNs considering the bandwidth guarantees, while minimizing the capacity reservation and the topological dispersion of the VPNs.

To minimize the bandwidth reservation, regardless which demand belongs to which VPN, is the same as routing multiple individual demands in the network. This is the classical minimum cost multicommodity flow problem. Linear programming will suffice for multicommodity flow if fractional (split) flows are permitted, however the multicommodity integral (unsplit) flow problem is NP-complete, even with only two commodities. This problem was extensively studied with emphasis on the approximation algorithms [Rad95, LMP+91, KARR90, Min89, LR88, Hu63].

In contrast with the path calculation in capacitated network, VPN topology design is not so widely studied [LGN+01, KEKAP02]. The VPN topology in most implementations has only two choices: fully-meshed or hub-and-spoke (star). In this thesis work I give an intermediate alternative that does not restrict the topology to be a tree, while still minimizing both topology and capacity jointly. The objective of the research is the topological design and route configuration of multiple overlapping

bandwidth guaranteed VPNs, which is an important issue regarding the costs. Encryption and security is handled by the upper communication layers, these aspects are not investigated.

By VPN design or configuration the route selection and the reservation of resources is meant, where the following conditions are assumed:

- bandwidth demands are static
- the network is capacitated
- multiple VPNs exist within a single physical network

The bandwidth demands between the VPN endpoints are static which is a proper assumption because creating a VPN is a long-term decision, and the "holding time" of the VPN can be several months or years, during this period the bandwidth requirements are fixed and not changing. The network is capacitated, i.e. capacity bounds limit the free bandwidth on the links. A given node in the network can be part of more than one VPN, i.e. VPNs could overlap.

The VPNs are formed by full mesh demand sets between VPN endpoints, i.e. all sites are interconnected. The service demands of VPNs are characterized by the bandwidth requirements of node-pairs (pipe-model).

In a physical network multiple different VPNs co-exist. The goal is to find the optimal configuration of these VPNs. The configuration of the VPNs is given with the routes between the VPN endpoints and the reserved capacities along the routes. The routes, that are determined with the design methods, are then set up in form of bandwidth guaranteed tunnels in the network that supports resource partitioning.

There are alternatives for the routes: they can be split or unsplit. The determination of splittable routes is much simpler problem than of unsplittable routes. Both alternative is investigated by the VPN design without protection.

The total cost of the VPNs consist of *administrative cost* and *bandwidth proportional* cost. The administrative cost is proportional with the numbers of used edges by a particular VPN, i.e. the topological dispersion of the VPN. If the topology of the VPNs is minimized, then the traffic is concentrated on few edges and the administrative cost will be minimized. The bandwidth proportional cost depends on the reserved capacity, therefore the other objective is to minimize the capacity reservation. These are the two objectives of the optimization that can produce opposite effects, but it is known that network design is always a trade-off among different objectives.

VPN users also require reliable connections therefore protection schemes are used to provide service guarantees. Path protection algorithms are studied widely [KL03] especially for optical networks [AQ00, DG01, HH04] together with wavelength assignment problem [ZOM03]. For the protected VPN configuration path based pro-active dedicated and shared protection methods are investigated. By the protected VPN configuration two disjoint routes – a primary and a backup – are sought. The routes

are not allowed to split in the protected VPN configurations. The objectives are the same as by the not protected VPN configuration, i.e. minimizing the capacity reservation and the topology of the VPNs.

3 Methodology of Research

The network is modelled as a graph, where the network nodes are represented by vertices and the network links are represented by edges. Each edge has an assigned value: its capacity. The endpoints of the VPNs are subsets of the physical network nodes. In the model all network nodes are potential VPN endpoints representing a host or a subnet behind it, i.e. any network node can be an edge node. The VPNs are defined with the traffic matrix that describes the bandwidth requirement between the endpoints of the VPN.

To calculate the global optimum an exact method, namely the integer linear programming was used. *Integer linear programs* were formulated for the different problems classes and ILP solver software (ILOG CPLEX 9.0 [cpl]) was applied to solve them.

The state space, in which the solver needs to find the optimal solution, was huge, therefore it could take very long time. However, heuristic methods can provide good-quality suboptimal solution, while the running time can be reduced significantly. One method, that was applied, is to *decompose the global ILP problem* into smaller problems and then solve them separately. Another method was to use *shortest path algorithms* to determine the paths one-by-one in each VPN, which can be considered also as a decomposition on the demand level. In both cases *ordering heuristics* were applied to determine the optimal order of subproblems and demands respectively.

In this thesis work integer linear programming problems are formulated, decomposition of global problems are described, and heuristic algorithms are presented for the optimal bandwidth guaranteed VPN configuration without and with protection in capacitated networks, where multiple VPNs co-exist. The parameter settings of the different methods are analysed to get the best solution. Performance metrics are defined to evaluate the capacity and topology objectives.

The methods are evaluated by simulation on test reference networks with generated VPN traffic patterns. Three backbone networks have been chosen that are based on real network topologies: NSFNet [nsf], COST-266 core network [IKM03], and a simplified version of the GÉANT Network [gea]. The network parameters can be seen in Table 1.

The results of the different methods are investigated and compared. The effectiveness of the heuristic methods are compared with the integer linear programming. The networks are relatively small, because I wanted to get results in reasonable time with the integer linear programming. The examinations for larger networks with the

Network	Nodes	Edges	Average edge degree	Diameter
NSFNet	13	19	2.923	3
COST266 core	16	23	2.875	5
GÉANT	18	30	3.333	5

Table 1: Selected Networks

heuristic algorithms can be found in my publications: for 80 nodes in **[C8, C9]**, for 20, 50, 100, 200 nodes in **[W1]**, for 28, 80, 137 nodes in **[J1]**. These papers show that the results of the heuristics algorithms are similar, independently of the network size.

The generated VPN traffic was the following: The number of VPNs usually does not depend on the network size, therefore three categories were created: 5, 10 and 15 was the number of VPNs. The size of the VPNs (number of VPN endpoints) is limited by the number of nodes in the network. Three categories were created again: small VPNs (number of endpoints ranges from 3 to 50% of the number of network nodes), large VPNs (number of endpoints is over 50% of the number of network nodes) and various size VPNs (from 3 to the number of network nodes).

The set of the tested configurations was the following:

- 3 networks
- number of VPNs (5, 10, 15)
- 3 VPN sizes
- generated bandwidth of demands (constant, 2 different uniform distribution, 2 different normal distribution)

The multiplication of these different types ($3 \cdot 3 \cdot 3 \cdot 5 = 135$) gives 135 test case, which includes 1,350 VPNs and 32,485 demands. So many test cases were evaluated in each design method with multiple, different settings.

4 New Results

To compare and evaluate the different design methods the following metrics have been defined. To evaluate the capacity objective the reserved capacity is determined relative to the available capacity in the empty network.

To evaluate the topology objective several metrics have been defined. For unprotected VPN design the number of tree topology VPNs are determined relative to the number of all VPNs. For protected VPN design this is not applicable because in protected VPNs each endpoint pair has two disjoint paths – a primary and a backup path – forming a cycle. Therefore two new topology related metrics were constructed the *VPN extension* and the *VPN node coverage*, that can be used for the unprotected VPN design as well.

Definition 1. *VPN extension is the number of VPN edges divided by the number of VPN nodes-1.*

In the unprotected VPN design, for a tree topology that contains only the VPN nodes the VPN extension is 1, because a tree contains nodes-1 edges. If the VPN extension is greater than 1, then either the VPN contains additional nodes because the VPN endpoints can not be connected directly, or if the VPN contains only VPN endpoint nodes, it is not a tree, or both. Although for protected VPNs the VPN extension is always greater than 1, it is a good measure for their topological dispersion too. The *VPN extension* metric is calculated for each VPN, the metric for the whole VPN configuration is the average of the individual VPN extension values.

Definition 2. *VPN node coverage is the number of nodes that belong to a VPN (i.e. the VPN has a path that goes trough that node or it is the endpoint of the VPN) divided by the number of physical network nodes.*

VPN node coverage specifies how many edges belong to a VPN relative to the total number of network nodes. The *VPN node coverage* metric is calculated for each VPN, the metric for the whole VPN configuration is the average of the individual VPN node coverage values.

If the node resources are partitioned among the different VPNs, it is an interesting issue, how many partitions have to be created, because fewer partitions means less overhead in the management system of the nodes. If the individual VPN node coverage values are summed, it gives the number that specifies how many partitions must be created on a node on average.

The following metrics were also used that were not directly evaluating the capacity and topology objectives. The average path length for all VPN demands expresses whether the paths follow the shortest possible path or they were diverted to longer detours. The running time of the algorithms were also measured. Comparing the running times in case of exact methods (ILP) shows the complexity of the given problem while in case of heuristics it shows the efficiency of the heuristic.

Some heuristic design methods cannot always solve the problems with the same constraints entirely as the exact methods. In these cases the constraints can be relaxed to get a full solution, or by keeping the original constraints the proportion of the partial solution can be measured. In the evaluation both ways were examined. In the latter case the granularity of the partial solution depends on the design method. With VPN based decomposition whole VPNs can fail, while with path based decomposition only paths can fail, which results in that some VPNs are not fully connected.

4.1 Bandwidth Guaranteed VPN Design Without Protection

THESIS 1: *Integer Linear Programming for Bandwidth Guaranteed VPN Design Without Protection*

THESIS 1.1: *I have formulated integer linear programs for unprotected VPN design with both splittable and unsplittable flows that minimizes the reserved capacity and the topological dispersion of VPNs jointly. ([**D1**] 3.1.2, 3.1.3; [**C4**, **C5**])*

The average time consumption depends on the balancing parameter (α) between the capacity and topology objectives both with splittable and unsplittable flows. Enforcing the topology minimization requires more computational time.

To create the objective function cost functions are assigned to the reserved capacity and to the number of virtual links, i.e. the "size" of the topology, namely the topological dispersion. The combined objective function of VPN optimization is the following:

$$\text{Cost(VPN)} = \alpha\ \text{Cost(reserved capacity)} + (1-\alpha)\ \text{Cost(number of virtual links)},$$

where α ($\alpha \in \mathbb{R}; 0 \leq \alpha \leq 1$) parameter balances between the two objectives.

To minimize the bandwidth reservation regardless which demand belongs to which VPN is the same as routing multiple individual demands in the network. This is the classical minimum cost multicommodity flow problem. If the flows are allowed to split linear programming suffices, however the multicommodity unsplittable flow problem is NP-complete, even with only two commodities.

The balance between the capacity and topology cost components is determined by the α parameter. Using this α parameter in the objective function enables that arbitrary balance can be evaluated. α is an input parameter in the optimization process. The actual setting of α is determined by the VPN service provider, that designs the VPNs. The particular value of α depends on the VPN service provider's technology, hardware and software devices, etc., and on the associated cost model with them.

In the objective function the cost of the reserved capacity and the cost of the number of virtually used edges differ significantly. The amount (number of units) of reserved bandwidth is typically greater even by orders of magnitude than the number of virtual links. To emphasize the topology objective the value of α parameter must be less then $\frac{1}{\max(\text{bandwidth})}$, where the maximum is taken on all demands. The maximum bandwidth in the test traffic input files were under 1000 unit, therefore $\alpha \leq 0.001$ places more emphasis on the topology objective in the simulations.

I analyzed which α settings are interesting for the optimization problem, therefore I examined the effects of the α parameter on the results. Table 2 shows the average values of the metrics calculated on the NSFNet traffic configurations. The result calculated in the other two networks showed the same tendencies, only with different

particular values. If the cost of one unit of bandwidth is equal with the cost of one virtual link, then $\alpha \to 0$ provides the minimal overall cost (denoted as solution value in the table), because the amount of reserved bandwidth is always more than the number of virtual edges, and α cannot be zero, because then the capacity objective would have been omitted from the objective function.

α	0.9	0.5	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
Solution value	31608	17456	3616.46	454.05	112.36	72.56	68.08	67.52	*67.71*
Time (sec)	0.36	0.61	1.29	9.86	1808.00	2722.16	2943.72	3021.72	3377.33
Capacity reservation (%)	59.63	59.63	59.63	61.76	68.87	70.47	*70.33*	70.87	71.11
Path length	2.22	2.22	2.22	2.30	2.53	2.59	*2.58*	2.59	2.60
Tree VPNs (%)	9.88	9.88	9.88	28.15	66.91	69.14	69.75	*67.90*	*65.68*
VPN extension	2.44	2.42	2.39	1.92	1.54	1.53	1.53	1.53	*1.54*
VPN node coverage (%)	72.17	71.98	71.47	63.14	57.37	*57.61*	*57.56*	*57.51*	*57.55*
Time limit reached					+	+	+	++	+++
Minimization effect	cap.	cap.	cap. & top.	cap. & top.	cap. & top.	cap. & top.	top.	top.	top.

Table 2: Effects of the α Parameter

In the ILP formulation the reserved capacity is represented by integer variables (supposing that the bandwidth requirements of the demands are integer multiples of a unit). However, whether a virtual link is used by a VPN or not, is represented by binary (0 or 1) variables, that makes the problem more complex. Thus, if the topology minimization is present in the objective function the solution time gets much longer.

Therefore, one hour time limit was set in the solver, to be able to evaluate the 135 test cases in reasonable time for each different α values.

The table shows that the overall cost reduction in the solution value is marginal if $\alpha \leq 10^{-5}$. The time consumption shows that the one hour time limit in the solver is reached in increasing number of the test cases if $\alpha \leq 10^{-3}$. This causes that several values in the right side of the table, written in italics, are not exactly following the deceasing or increasing order of the metrics, because they are from a suboptimal solution.

If $\alpha \geq 10^{-1}$, the reserved capacity, the path length, and the ratio of tree VPNs are minimal, and are equal for the different α values. Although the VPN extension and the VPN node coverage is decreasing, this is only marginal.

If $\alpha \leq 10^{-4}$, the capacity and topology metrics also reach their extreme values, and decreasing of the α parameter does not cause further significant change in the values.

Therefore, $\alpha \geq 10^{-1}$ realizes mainly the capacity minimization, $\alpha \leq 10^{-4}$ realizes mainly the topology minimization, and $10^{-4} \leq \alpha \leq 10^{-1}$ realizes jointly the capacity and topology minimization. Thus, the design methods always will be evaluated with $10^{-4} \leq \alpha \leq 10^{-1}$ parameter values.

Table 3 shows the differences in the average time consumption between the splittable and unsplittable flow problems averaged over all test configurations (the tests were run on a 2 GHz PC). Lower α values are not presented because the 1 hour time limit, that was set in the solver, was reached in significant number of the test cases.

α	Average time consumption with splittable flows	Average time consumption with unsplittable flows
0.9	1.29 sec	105.10 sec
0.1	5.75 sec	80.15 sec
0.01	75.96 sec	281.06 sec

Table 3: Average Time Consumption of Splittable and Unsplittable Flow Problems

THESIS 1.2: *If flow splitting is allowed, only relative small fraction of the paths will be split, on average up to approximately 6%. However, split flows are present in most of the VPN configurations. The topology minimization reduces the number of split flows and also the number of cases, where split flows are present (from 97% to 84%). ([**D1**] 3.1.2)*

For VPN design with unsplittable flows around 10% of the cases cannot be solved that could be solved with splittable flows. However, adding 10% extra capacity results in solving the remaining 10%.

The simulations have shown that splitting is exploited in most of the test cases, however only small part of the VPN paths will be split. The lower time consumption and the small number of split flows suggests a practical way if unsplit flows are required: with the appropriate capacity extension only the split flows need to be recalculated if unsplittable flows are required.

The VPN traffic was generated to reach the highest possible network load. The maximal demand bandwidths have been determined for a given bandwidth distribution that is still feasible and the solution gives the minimal bandwidth reservation with splittable flows. Therefore, if the flows are not allowed to split, not all traffic configuration could be solved.

Another observation was, that, whether the VPN traffic configuration could be solved with unsplittable flows or not, does not depend individually on the number of split flows in the splittable solution. Except, that if there is a solution with not split flows in the splittable case, then this is also a solution in the unsplittable case.

THESIS 1.3: *The topology minimization can be achieved only to a limited degree without increased capacity reservation. In an intermediate balance between the capacity and topology objectives the capacity reservation increase is moderate, while the VPN topology metrics indicate significant topology reduction. Emphasizing the topology objective further improves the topology reduction according to the metrics, however it requires considerably more capacity.*

These statements are valid both for the splittable and unsplittable flow problems. Figures 1 and 2 illustrate the result for the VPN design with splittable flows. For each metric (see page 6) the three groups represent the three test networks, and inside the groups the results corresponding to different α settings are shown. If $\alpha \geq 0.1$ the capacity reservation is constant, the intermediate balance is at $\alpha = 0.01$, and if $\alpha \leq 0.001$ the capacity reservation is increased approximately with 10%.

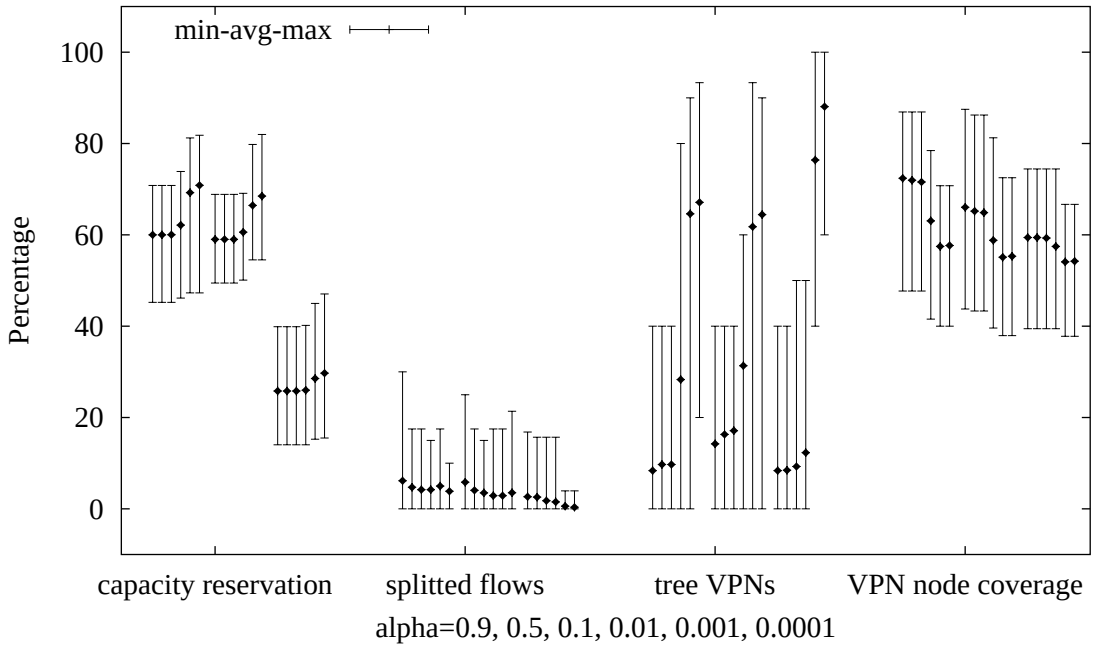


Figure 1: Minimizing Capacity and Topology with Splittable Flows – Relative Metrics

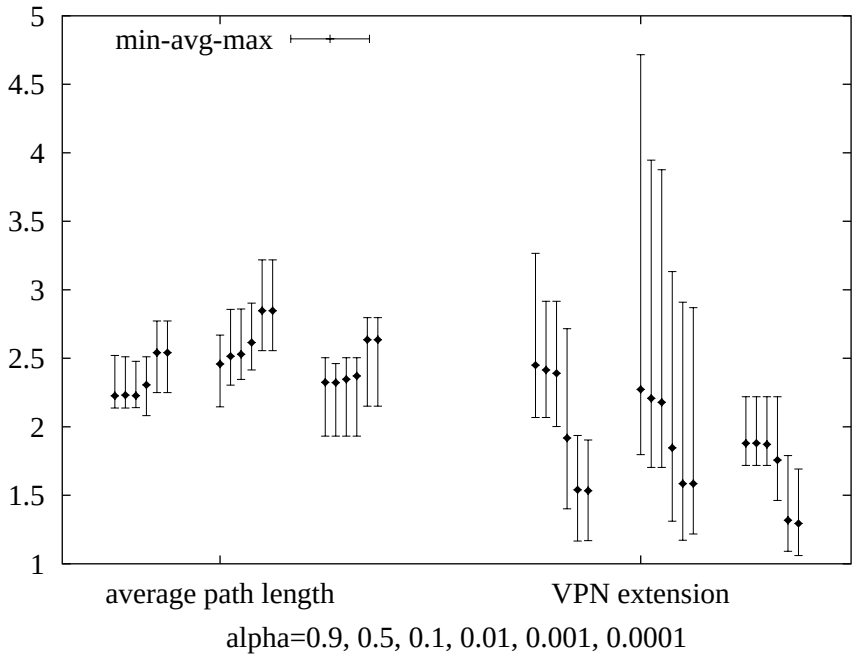


Figure 2: Minimizing Capacity and Topology with Splittable Flows – Absolute Metrics

THESIS 2: *Heuristic Methods for Bandwidth Guaranteed VPN Design Without Protection*

Heuristic methods might not always find the best solution, however it is guaranteed to find a suitable solution in reasonable time. With the global optimization technique, where all VPNs with all their traffic demands are optimized together, the global optimum is guaranteed whenever it exists and the solver can provide it in acceptable time. However, this problem can be quite complex and time consuming as in Thesis 1 with enforced topology minimization the 1 hour time limit was reached in most of the cases. Therefore, I have worked out three heuristic methods for the VPN design without protection.

The heuristic methods are based on decomposition. A decomposed subproblem is to determine one VPN or one path. After a subproblem is solved, the available free capacity is reduced in the network according to the solution, thus the next subproblem will have less resources. Previously calculated VPNs or paths can reserve so much capacity on some links that the subproblems which are computed later can get unfeasible. Therefore, in part of the traffic configurations not all VPNs or VPN routes can be determined with the heuristics, i.e. only a *partial* solution is obtained. If all VPNs, i.e. all VPN routes can be determined then the solution is *complete*. To increase the number of complete solutions the capacity of the links has been enlarged, i.e. *capacity extension* was added to each link.

THESIS 2.1: *I have proposed a VPN based decomposition of the global integer linear programming problem with score based ordering heuristics for VPN design without protection. ([**D1**] 3.2.1; [**C7**])*

Emphasizing the topology objective in VPN based decomposition results in increased number of partial solutions. However, this affects only small ratio of VPNs. To achieve more complete solutions with topology minimization the relaxation of capacity bounds is required. Adding a relative small capacity extension drastically reduces the ratio of partial solutions.

If the VPN based decomposed heuristic is able to give a complete solution for the VPN design, then the results compared to the global ILP are within 5% difference.

To reduce the computational complexity, the global problem has to be decomposed into smaller subproblems that could be solved separately in significantly shorter time. The global solution is assembled from the results of the solutions of the subproblems. The important gain is the reduced running time in return for the sub-optimality. The straightforward decomposition is done according to the VPNs. The number of subproblems is the number of VPNs, these smaller subproblems are going to be solved one-by-one. Consequently, the complexity is reduced because instead of the whole demand set only the appropriate VPN's demand set is included in the subproblems, which yields significantly less variables and constraints, thus reducing the search space.

The integer linear programming subproblems that are formulated separately for each VPN are solved one after another. After a solution for a VPN is ready, the available bandwidth is reduced on the links according to the solution, thus the available free capacity will be lessen in the next optimization. Therefore, the order in which the VPNs are calculated matters in determining the actual configuration. VPNs that are calculated early in the process have more resources available than those calculated later because previously calculated VPNs or paths can consume some critical network resources and can hinder new demands – and therefore VPNs – getting accommodated.

The goal of the demand ordering is to achieve more complete solutions. This is similar to packing problems (knapsack, bin packing) [GJ79]. In the packing problems the volume and the cost of the objects are known. However, when routing a demand the reserved capacity in the network depends on the selected route. Therefore, I proposed the following ordering heuristics: each demand gets a score according to the bandwidth requirement of the demand and the distance between the source and destination nodes. This score estimates the "volume" of the path. These demand scores are summed by VPNs and this will be the score of the VPN. This gives the order in which the VPNs are optimized.

The scoring system is based on the bandwidth requirement of the demand and the distance between the source and destination nodes. The question is whether to

prefer high or low bandwidth demands, near or far endpoints? The basic concept was to prefer high bandwidth demands with near endpoints. However, the test runs have shown, that for splittable flows this is not the proper preference mode. Although the difference is within 5%, preferring demands with low bandwidth and with far endpoints results in more full solutions. This is due to the ability to split flows, because the demands, that are going to be routed later in the process, can split to several small bandwidth flows, and therefore the capacity bounds are not so restrictive as in the case of unsplittable flows.

Figure 3 shows the tendencies in the ratio of partial solutions according to the capacity–topology balancing and the relaxation of capacity bounds. Emphasizing the topology objective with decreasing α increases the ratio of partial solutions from 25% to 80% with splittable flows and from 55% to 88% with unsplittable flows, while the ratio of not configured VPNs increases from 3% to 16% with splittable flows and from 7% to 17% with unsplittable flows. Adding only 10% of extra capacity the ratio of partial solutions is reduced to 30% with splittable flows and to 38% with unsplittable flows. It can be seen also that 50% capacity extension provides complete solution for the extreme test cases.

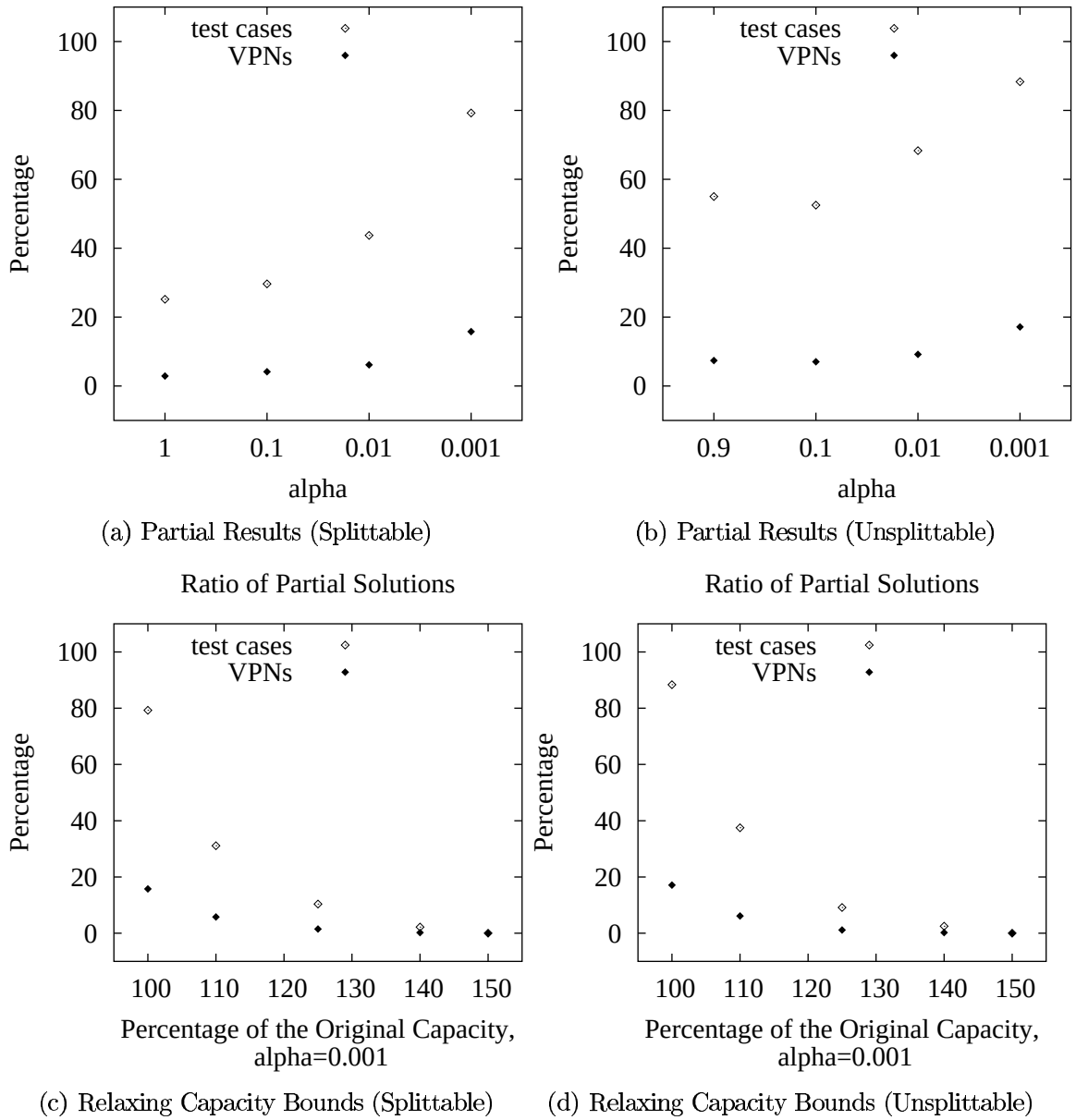
The sub-optimality of the heuristic is indicated by 5% difference between the results, if the heuristic provides complete solution with the original capacity bounds. However, it must be noted, that to get complete solution, capacity bound relaxation might be needed, that also lessens the quality of the solution. The important gain is the reduced time consumption of the VPN based decomposition which is lower in an order of magnitude compared to the global ILP problem. It remains under 1 min even with the emphasized topology minimization.

THESIS 2.2: *I have proposed a path based decomposition with score based ordering heuristics for VPN design without protection to realize capacity and topology minimization jointly. ([**D1**] 3.2.2; [**C7**])*

The path based decomposition with score based ordering heuristics for VPN design without protection is able to approximate the intermediate level ($\alpha = 0.01$) of the capacity–topology minimization compared to the results obtained with ILP.

Adding a small amount of extra capacity drastically decreases the ratio of partial solutions and also improves the topology minimization.

A fast and simple method is to calculate the paths of the VPNs one-by-one. To find a path between a source and a destination Dijkstra's shortest path algorithm was used [Dij59] which finds one unsplit flow. Dijkstra's algorithm finds the shortest path between two vertices in a directed or undirected non-negative weighted graph in polynomial time.



(a) Partial Results (Splittable)

(b) Partial Results (Unsplittable)

(c) Relaxing Capacity Bounds (Splittable)

(d) Relaxing Capacity Bounds (Unsplittable)

Figure 3: VPN Based Heuristics with Splittable and Unsplittable Flows

Setting the weights according to the appropriate objectives determines which path will be found by Dijkstra's algorithm. The weights of the links are responsible for:

- ensuring the bandwidth guarantee
- minimizing the capacity reservation
- minimizing the topology

The bandwidth guarantee is realized by excluding the links that do not have enough free capacity to accommodate the demand, i.e. setting the weights of those links to infinite. If the goal is the capacity objective, i.e. to minimize the capacity reservation, the weights of the edges are set to the bandwidth to be reserved on that edge if the selected demand gets accommodated on that edge. If the main goal is the topology objective, i.e. to minimize the number of virtual links, then the already used edges get lower weight than the unused ones. This way, the new paths prefer to follow the already established ones, not allowing the dispersion of the paths. The newly established paths alter the set of already used edges, therefore the weighting of links is modified in each step.

The combination of the capacity and topology objectives are expressed in the following weighting scheme. The weight of an edge is the currently processed demand's bandwidth, if a path of a demand already uses that edge from the same VPN. This base weight is responsible for the capacity objective. Otherwise, the weight is the bandwidth of the currently processed demand plus an offset, which is responsible for the topology objective. This additional weight tries to keep the paths along the already established ones. This offset was chosen to be the average demand bandwidth multiplied by a parameter, which was set to 10 in the calculations, i.e. the topology minimization was enforced.

The effect of the offset is the following. If the multiplier parameter is e.g. 10 and we consider a demand with average bandwidth, the cost of an 11 hops long path consisting of already used edges is equal to a one hop long path (i.e. a direct "shortcut" edge) which is not yet used.

Therefore for an average size demand, if there is an available path containing already used edges and it is not too long, it will be chosen, if it is too long, then a new, not yet used shortcut path will be chosen. Whether a path is regarded as too long or not depends on the multiplier parameter. A demand with bandwidth above the average bandwidth is less likely to follow the already used edges, while a demand with bandwidth below the average bandwidth is more likely to do so. Thus, the already used edges are preferred, but demands with over average bandwidth are avoided to route on too long paths.

The scoring system is similar to the one used at the VPN based heuristics. Each demand gets a score according to the bandwidth requirement of the demand and the distance between the source and destination nodes. Since in this method

the demands are individually scored and routed, the scoring has greater influence to the ratio of partial solutions. After investigating the scoring system variants ([**D1**] 3.2.2), to route as many demands as possible is satisfied with the high bandwidth – near endpoints preference with 0.75 distance to bandwidth weighting ratio: $\text{Score}_{\text{demand}} = \frac{3}{4}\left(1 - \frac{d}{d_{\max}}\right) + \frac{1}{4}\frac{b}{b_{\max}}$, where b is the bandwidth of the demand, and d is the distance between the endpoints, and the maximum is taken on all demands. It must be noted, that even if the ratio of full solutions is under 17.5%, the ratio of not routed demands is under 6%, but this 6% of demands are distributed among the VPNs in such a way, that only less than 17.5% is not affected.

To compare the effectiveness of the heuristic the results were compared to the results of the ILP solution. Table 4 shows in which region the results of the heuristic are according to the capacity–topology balancing parameter α in ILP formulation. In the columns the results obtained by solving the ILP with the corresponding α setting are displayed. With decreasing α the results of ILP for metrics tree VPNs, time consumption, capacity reservation and average path length are increasing, while for VPN extension and VPN node coverage are decreasing. The results of the heuristic are placed between the appropriate two values of the ILP solution highlighted with bold frames. The topology related metrics (tree VPNs, VPN extension, VPN node coverage) correspond to $\alpha \approx 0.01$ in the ILP solution. However, the average time consumption is very low. The capacity reservation and the average path length is higher than in the ILP solution with emphasized topology objective ($\alpha = 0.001$). Compared to the ILP solution with $\alpha = 0.01$ the differences are within 5%, except for the average path length, that is higher with 0.5.

α		0.9		0.1		0.01		0.001	
tree VPNs (%)		8.56		10.53	24.97	30.77		66.35	
VPN extension		2.20		2.14	1.89	1.84		1.47	
VPN node coverage (%)		66.92		66.00	64.24	60.50		55.98	
Time consumption (sec)	0.10	32.11		33.76		156.56		1993.79	
Capacity reservation (%)		48.21		48.21		49.56		54.11	55.63
Average path length		2.39		2.39		2.45		2.72	2.92

Table 4: Path Based Heuristic Compared to ILP

The relaxation of the capacity bounds improves the results of the heuristic. As by the VPN based heuristic, adding a small amount of extra capacity drastically decreases the ratio of partial solutions and also improves the topology minimization. Adding 50% extra capacity can achieve the emphasis on the topology objective, the results are nearly as good as the ILP solution with $\alpha = 0.001$, while the time consumption is still around 0.1 sec.

THESIS 2.3: *I have proposed a Simulated Allocation scheme for VPN design without protection to realize capacity and topology minimization jointly. ([**D1**] 3.2.2; [**C9, J1**])*

The emphasized topology minimization with Simulated Allocation scheme for VPN design without protection is able to outperform the intermediate level ($\alpha = 0.01$) of the capacity–topology minimization regarding the topology metrics.

Simulated Allocation allows more options to test in the search space by not only reserving but also deleting paths in a converging way. Therefore, more variations are examined, which enhances the quality of the result but the running time is also longer.

Simulated Allocation was proposed by Michal Pióro [PSG+00]. The algorithm was adapted to the VPN design and enhanced with the score based probability based random selection. The demand for routing is chosen in a probabilistic way from the set of the not yet routed demands. The probability of choosing a demand is proportional to its score. The score of the demand is calculated in the following way (the maximum is taken over all demands): $\text{Score}_{\text{demand}} = \frac{1}{2}\left(1 - \frac{d}{d_{\max}}\right) + \frac{1}{2}\frac{b}{b_{\max}}$, where b denotes the bandwidth of the demand, and d denotes the distance of its endpoints. Thus, high bandwidth demands with near endpoints are preferred as previously by the path based heuristic. The test runs have shown that with simulated allocation a balanced emphasis between the distance and bandwidth based scores performs better (0.5–0.5 instead of 0.75–0.25), because of the reserving–freeing steps.

The ratio of complete solutions is influenced by two parameters: the number of iterations and the degree of capacity relaxation. Table 5 shows the ratio of complete results and the time consumption by increasing the iteration count with the initial capacity bounds.

Increasing the iteration count increases the running time linearly, however, even with 1,000,000 iteration, the ratio of complete solutions is still not 100%. Therefore the capacity bounds were relaxed. The tests have shown that only 5% of additional capacity and 10,000 iteration is sufficient to achieve 100% of complete solutions (see Table 6).

Iteration count	Ratio of complete solutions	Average time consumption (sec)
1,000	61.67%	2.7
10,000	81.67%	24
100,000	91.67%	257
1,000,000	94.17%	2436

Table 5: Increasing the Iteration Count in Simulated Allocation

Additional capacity (%)	Iteration count	Ratio of complete solutions (%)
10	1,000	98.33
10	10,000	100
5	10,000	100
3	10,000	96.67

Table 6: Relaxing the Capacity Bounds for Simulated Allocation

Table 7 shows the quality of the results of the Simulated Allocation with emphasis on the topology minimization compared to the ILP solution regarding the topology related metrics. Most of the values are between α 0.01 and 0.001, that means the topology objective is emphasized but not as much as with the ILP with $\alpha = 0.001$. The table shows, that in shorter time (75 sec vs. 156 sec) approximately with 10% better values can be reached regarding the ratio of tree VPNs and VPN extension than with ILP with $\alpha = 0.01$. The VPN node coverage is only slightly worse than the ILP with $\alpha = 0.01$. The capacity reservation and the average path length is higher, but only with approximately +5% and +0.36 respectively.

Comparing this table with Table 4 it can be seen that in return for a higher time consumption all other metrics are considerably better with Simulated Allocation than the Path Based Heuristic with Score Based Ordering.

α		0.9		0.1		0.01		0.001	
tree VPNs (%)		8.56		10.53		30.77	41.37	66.35	
VPN extension		2.20		2.14		1.84	1.71	1.47	
VPN node coverage (%)		66.92		66.00	61.02	60.50		55.98	
Time consumption (sec)		32.11		33.76	75.28	156.56		1993.79	
Capacity reservation (%)		48.21		48.21		49.56	54.01	54.11	
Average path length		2.39		2.39		2.45		2.72	2.81

Table 7: Simulated Allocation Compared to ILP

4.2 Bandwidth Guaranteed VPN Design With Protection

To ensure reliability – which is an important value-added factor for a VPN service – the design must be prepared for failures. Protection reserves resources in advance to use them in case of failure. Protection is pre-planned, therefore it is used with more static traffic patterns, like VPNs. Because the demands are known in advance the protection can be planned in advance as well.

Path protection reserves disjoint backup paths for primary paths. The backup path can be link or node disjoint with the primary path. In the node disjoint case the primary and backup paths are link disjoint as well. Path protection assumes that the primary path is unsplit, which is typical in telecommunication networks, and calculates also an unsplit backup path.

The objective is the joint capacity and topology minimization as well as by unprotected VPN design. However, because the primary paths and backup paths are disjoint, the topology objective cannot result in tree topology, therefore this metric is omitted here. On the other hand, compacting the protected topology still reduces the expansion of the VPNs.

THESIS 3: *Integer Linear Programming for Bandwidth Guaranteed VPN Design With Protection*

THESIS 3.1: *I have formulated the integer linear programs for the link and node disjoint dedicated protection for protected VPN design. ([**D1**] 4.1.1, 4.1.2; [**C5**])*

By the same traffic volume, more demands with smaller bandwidth require more capacity for protection than fewer demands with larger bandwidth requirements.

The integer linear programming problem gets more complex in case of protected VPN design, because additional variables and constraints are presented for the protection paths, and all these variables are binary.

To guarantee bandwidth for the backup paths the capacity bounds need to be relaxed. The question is how much extra capacity is required to achieve that all primary paths have a backup path with the same bandwidth guarantees. If the capacity bounds were doubled, only 40% of the test cases was feasible. The test configurations in which are many VPNs and/or the VPNs are various sizes (from 3 endpoints to the number of nodes in the physical network) require more capacity for protection. Because the traffic volume was approximately the same in all traffic files, many VPNs means more demands with smaller bandwidth requirements. Thus, this kind of traffic configurations require more capacity for protection compared to fewer demands with larger bandwidth requirements. To be able to solve almost all test cases (more than 97%) the capacity bounds were increased to 2.5 times of the initial.

THESIS 3.2: *Including dedicated protection to the VPN configuration requires 2.5 times more capacity. Emphasizing the topology objective causes increase in the backup capacity reservation and average backup length, while the primary capacity reservation and average primary path length is almost constant.*

In Figure 4 the primary and backup capacity reservation is depicted separately, and also the total capacity reservation. It can be seen that pushing the topology objective (decreasing α) affects mainly the backup capacity reservation, the primary capacity reservation is almost constant. The VPN node coverage is in the 65–90% interval instead of the unprotected configuration's 55–75%. This indicates, that to

realize two disjoint paths between each VPN endpoints, the paths have to go through 10–15% more nodes in the network. The price for the VPN node coverage, (i.e. also the VPN topology) minimization is the increased capacity reservation. The figure shows that a good compromise is the $\alpha = 0.01$ setting, where the capacity reservation increase is minimal while the VPN node coverage is definitely reduced.

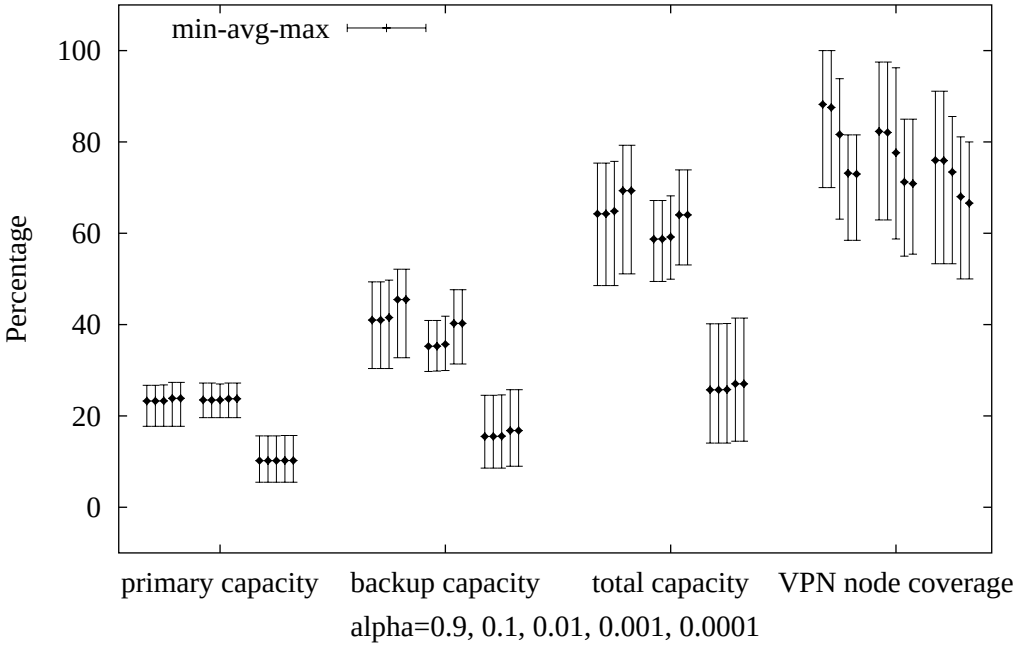


Figure 4: Link Disjoint Dedicated Protection with ILP – Relative Metrics

The result for the paths (see Figure 5) are similar to the capacity reservation results. The backup path length increase is more significant than the primary path length by enforcing the topology objective, however, there is a certain increase also in the primary path length. The average total path length is higher (approximately 3–4) than that of the not protected configuration (approximately 2.25–3). The VPN extension is about twice as that of the not protected VPN configuration (2.5–1.25 vs. 3.5–2.25), but it can be reduced as well as by the not protected VPNs (see Figures 2 and 5). Figure 5 also illustrates that a good compromise is the $\alpha = 0.01$ setting.

The average time consumption was dependent on parameter α. For $\alpha \geq 0.01$ the average time consumption was 1.5–2 minutes, for $\alpha = 0.001$ around 14 minutes, and for $\alpha = 0.0001$ it was around 47 minutes. In the last two cases there were significant number of test cases that has reached the one hour time limit. This indicates again that enforcing the topology objective requires more computation, at least one order of magnitude higher.

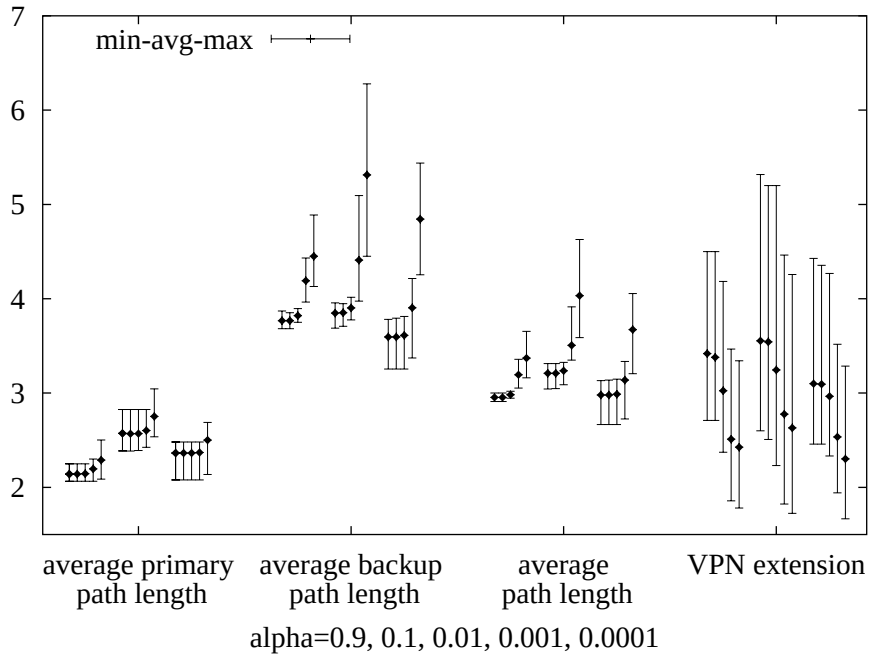


Figure 5: Link Disjoint Dedicated Protection with ILP – Absolute Metrics

THESIS 3.3: *The capacity and topology metrics are approximately equal by the configuration obtained by* node disjoint *dedicated VPN protection design and by the configuration obtained by the* link disjoint *dedicated VPN protection design.*

The simulations have shown that there are only very small differences (maximum 0.7% by metrics in percentage and maximum 0.05 by absolute metrics) between the results of link and node disjoint configurations. Many of the primary-backup path pairs in the configuration obtained by link disjoint design are also node disjoint.

Because of this, in the following only the *link disjoint* VPN design is investigated with the heuristics.

THESIS 4: *Heuristic Methods for Bandwidth Guaranteed VPN Design With Protection*

THESIS 4.1: *I have proposed a VPN based decomposition of the global integer linear programming problem with score based ordering heuristics for VPN design with protection. ([**D1**] 4.2.1; [**C7**])*

If the VPN based heuristic is able to provide complete solution it is very close to the global optimum and the time consumption is significantly lower. A small amount of capacity extension significantly decreases the ratio of partial solutions and improves the quality of the solution.

The integer linear programming problem is divided to VPNs to reduce the complexity. The subproblems are solved one-by-one according to the score assigned by the scoring system. By the test runs, the best performing scoring system was found to be the same as by the unprotected VPN design.

To solve the dedicated protection VPN design with ILP, the capacity bounds that had been used for the unprotected VPN design, had to be increased. The initial capacity bounds used for unprotected VPN design is the starting point, this is marked as 100%. The initial capacity bounds for the protected VPN design methods is 2.5 times the capacity bounds used at unprotected VPN design, marked as 250% in the followings. For protected VPN design this 250% will be considered as the initial capacity bounds.

The difference between the results of ILP and VPN based heuristic is by the metrics in percentage within 2% and by the metrics with absolute values within 0.1, i.e. if the VPN based heuristic is able to provide complete solution, it is very close to the global optimum and the time consumption is lower: approximately 1-1.5 min instead of several minutes.

The VPN based heuristic does not provide complete solution in all cases for the protected VPN design with the initial capacity bounds, similar to the heuristics for the unprotected VPN design. The ratio of partial solutions and also the ratio of not planned VPNs depends on the α parameter. To reduce the ratio of partial solutions the capacity bounds need to be relaxed again. The behavior is similar as depicted in Figure 3 for the unprotected VPN design: a small capacity extension significantly decreases the ratio of partial solutions, and capacity extension from 250% to 375% makes also the solutions of the extreme test cases complete.

THESIS 4.2: *I have proposed two path based heuristics for VPN design with protection that incorporate the score based ordering. The first one is based on Dijkstra's algorithm, the other one is based on Suurballe's algorithm. ([**D1**] 4.2.2; [**C7, C8, W1**])*

Comparing the two methods, Suurballe's algorithm yields better results, but the difference is not significant.

The topology minimization quality is limited, it can be realized only to a modest degree (corresponding to $\alpha = 0.1$).

In path protection two link disjoint paths must be determined for a demand. The path based heuristics search primary and backup path with shortest path algorithms in polynomial time.

Dijkstra's algorithm is used to route not only the primary paths but the backup paths too. At first, the primary path is calculated with Dijkstra's shortest path algorithm. Then, the links of the primary path are temporarily deleted from the network to ensure link disjointness and Dijkstra's algorithm is applied again. This solution gives the shortest path for the primary path. The backup path will be also the shortest possible path, but because the primary path is deleted, it can be much

longer. A drawback of this method is that deleting the edges of the primary path can prevent to find the backup path.

Suurballe's algorithm [ST84] searches the shortest pair of edge disjoint paths simultaneously. Therefore, the result is the shortest regarding together the primary and backup paths. Thus, it differs from the result of Double Dijkstra's algorithm. From the two paths the not longer will be the primary, and the other the backup.

By the test runs, the best performing scoring system was found to be the same as by the VPN design without protection: $\text{Score}_{\text{demand}} = \frac{3}{4}\left(1 - \frac{d}{d_{\max}}\right) + \frac{1}{4}\frac{b}{b_{\max}}$. The only difference is that both the primary and the backup paths are calculated for a selected demand.

The initial capacity for the protected configurations (250% of the not protected capacity) was not enough to solve all configurations completely by the heuristic. The ratio of complete solutions is depending on whether topology minimization is forced or not. However, because of the failure granularity (only demands instead of whole VPNs) even if the ratio of complete solutions is only 25–30%, the ratio of not routed demands is under 5–7%.

The ratio of complete solutions is increasing with the capacity bound relaxation, however, it could reach 100% only with emphasis on the capacity objective by 150% capacity relaxation. Comparing the two methods, Suurballe's algorithm yields somewhat better results, i.e. more complete solutions and less not routed demands, but the difference is not significant.

Tables 8 and 9 show the results of the heuristics compared to the results of the integer linear programming solution based on the α capacity–topology balancing parameter. The results of the heuristics with enforced topology minimization are in the bold frames. As it can be seen from the tables, the topology related metrics (VPN extension and VPN node coverage) the topology minimization realized only to a modest degree falling close to the values obtained by $\alpha = 0.1$ with ILP, i.e. the heuristics with the initial capacity bounds are not able to realize the compacting of the topology.

α		0.9		0.1		0.01		0.001		0.0001	
Primary capacity		18.88		18.88		18.90		19.18	19.98	20.14	
Backup capacity		30.34		30.36		30.69	32.74	33.92		38.86	
Total capacity		49.23		49.24		49.59	52.73	53.10		59.00	
Primary path length		2.36		2.36		2.36		2.39		2.52	2.55
Backup path length		3.74		3.74		3.78		4.17	4.19	4.88	
Path length		3.05		3.05		3.07		3.28	3.37	3.70	
VPN extension		3.36		3.34	3.25	3.08		2.63		2.45	
VPN node coverage		82.04		81.72	80.11	77.47		70.75		70.09	
Time consumption	1 sec	1.5-2 min		1.5-2 min		1.5-2 min		14 min		47 min	

Table 8: Double Dijkstra's Algorithm Compared to ILP Solution

α		0.9		0.1		0.01		0.001		0.0001	
Primary capacity		18.88		18.88		18.90		19.18		20.14	20.60
Backup capacity		30.34		30.36		30.69	32.75	33.92		38.86	
Total capacity		49.23		49.24		49.59		53.10	53.35	59.00	
Primary path length		2.36		2.36		2.36		2.39		2.52	2.54
Backup path length		3.74		3.74		3.78		4.17	4.20	4.88	
Path length		3.05		3.05		3.07		3.28	3.35	3.70	
VPN extension		3.36		3.34	3.21	3.08		2.63		2.45	
VPN node coverage		82.04		81.72	80.03	77.47		70.75		70.09	
Time consumption	0.25 sec	1.5-2 min		1.5-2 min		1.5-2 min		14 min		47 min	

Table 9: Suurballe's Algorithm Compared to ILP Solution

This comparison also shows that Suurballe's algorithm is slightly better, since the VPN extension is lower. Even when the topology is not minimized, the capacity reservation and the average path length is high, as by the ILP solution with emphasized topology objective. However, it must be noted that the difference between the ILP solution with emphasizing the capacity objective ($\alpha = 0.9$) and the results of the heuristics is maximum around 4% at the capacity reservation and 0.3 at the path length. Altogether, the heuristics are not able to achieve the enforced topology minimization, but the capacity objective is fairly approximated in at least one order of magnitude shorter time (seconds instead of minutes).

To improve the topology related metrics the capacity bounds need to be relaxed as by the unprotected VPN design. If the capacity bounds are relaxed to 150% of the initial values, the topology metrics (VPN extension and VPN node coverage) get closer to the ILP solution, but the difference is higher than this method has achieved by the unprotected case.

Because Suurballe's algorithm searches the shortest pair of edge disjoint paths the total backup capacity and total path length are slightly lower than the result of Dijkstra's algorithm, however as a side effect of this, the primary capacity and path length are higher and the backup capacity and backup path length are lower for the results obtained by Dijkstra's algorithm than the results obtained by Suurballe's algorithm. Thus, these results are in correspondence with the theoretical expectations coming form the definitions of the two algorithms.

THESIS 4.3: *I have proposed a Simulated Allocation based heuristics for VPN design with protection with dedicated and shared protection. ([**D1**] 4.2.2; [**J1, C8, C9**])*

With Simulated Allocation the topology minimization is moderate (corresponding to approximately $\alpha = 0.01$), however the time consumption remains lower than the ILP solver.

Shared protection significantly reduces the capacity reservation even with enforced topology minimization objective, where the topology metrics are also better than the results of Simulated Allocation with dedicated protection.