Shared Protection of Virtual Private Networks with Heuristic Methods

Péter Hegyi, Markosz Maliosz, Ákos Ladányi, Tibor Cinkler
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
Magyar tudósok krt. 2., Budapest 1117, Hungary
E-mail: hegyi@tmit.bme.hu, maliosz@ttt-atm.tmit.bme.hu,
ladanyi@tmit.bme.hu, cinkler@ttt-atm.tmit.bme.hu

Abstract

In our paper we analysed algorithms that seek the optimal routing configuration in backbone networks with capacity constraints. We investigated a special case when multiple full mesh demand sets (forming Virtual Private Networks (VPNs)) have to be built.

We examined the pro-active path based shared protection scheme and investigated heuristic algorithms to calculate the paths.

We analysed two different shared protection methods, and we also made comparisons between versions and objective types of the two algorithms. The first algorithm applies Dijkstra's shortest path calculation, while the second algorithm is based on local edge search. In the analysis and comparison numerical results are presented from calculations on sample networks.

1. Introduction

Virtual Private Network (VPN) is an emerging topic in recent network service provisioning. A VPN consists of several geographically separated sites that are connected to a service provider's network. VPNs support the communication requirements of a closed group of users with special handling of privacy and security, while give a cheaper alternative compared to leased lines. VPNs, in fact, are networks on networks: they seem to be real private networks for their users, but actually they are realised over the network of one or more providers.

In practice, the sites that use the VPN are connected to the provider's network through edge-routers [1]. This means, that if we treat the network as a graph, then during VPN design we can assume that a VPN is defined by demands between the pairs of points, where a point can represent a whole campus for example. In our model we suppose that the bandwidth of the demands is constant. This model can be used by a service provider for long term VPNs [2].

To ensure reliability the design must be prepared for failures. There are several methods to protect the VPNs. One way – for example – is to implement a version of Short Leap Shared Protection (SLSP) algorithm. This means that we break the working paths

into overlapping segments, and we calculate the protection for these segments [3]. One interesting solution of SLSP is to use the Cascaded Diverse Routing algorithm together with the Iterative Two Step Approach algorithm. This solution has two alternative paths for each segment, and when a demand arrives, it selects one of the two. This method is used online, i.e. it is dynamic [4].

There are also offline methods that use pre-calculated protection paths. The dedicated protection – where for the protection paths the full demand bandwidth is reserved – is a bit wasting solution. Since failures are rare, we can assume that in the same time only one link of the network is out of order. So we can decrease the bandwidth utilisation using shared protection [5]. In this case the protection paths can share their reserved bandwidth in the following manner. On an edge it is enough to reserve as much capacity, as the bandwidth of the biggest demand that we protect on it. However, if several protection paths use an edge and their corresponding working paths have common edges, they cannot share the protection bandwidth, thus we have to count the sum of these bandwidths.

2. The Methods

Generally it is true, that if we use just few edges, and we reserve much capacity on them, we pay to the provider less, than in that case, when we use more edges and we reserve small capacities on them. The reason for that is that for the providers it is worth to let higher capacities, because e.g., if the provider fritters away its bandwidth, it will have more administrative work to do, than in the case when it lets out the whole bandwidth for just one customer.

For the provider's point of view the statement is also true. By building our physical network it is worth to build a link with higher capacity, because the cost ratio is less at higher bandwidth links since the constant costs (digging the trench, cable cost, etc.) are the same.

It is also true, that when we use just a few edges and we reserve much capacity on these edges, we can share the protection better, than in that case, when we use many edges, with small capacities reserved on them.

We used Dijkstra's algorithm to route the working paths in both methods. The protection algorithms are presented in [6]. Here we just give a short summary.

2.1. Protection Algorithms

2.1.1. Two Times Dijkstra's (TTD) Algorithm

In this method we used Dijkstra's algorithm to route not only the working paths but the protection paths too. This is similar to the method discussed in [7].

Before we can run Dijkstra's algorithm we have to count the capacity available for protection on each edge taking into account the *sharing*. If the available capacity of the edge is not enough to accommodate the protection path or the edge is part of the working path, it is excluded from the edge set when running Dijkstra's algorithm.

2.1.2. Frequency Set (FS) Algorithm

The aim of this algorithm is to use as few new edges for protection as possible. The algorithm constructs the protection path step-by-step. It starts from the source node of the demand and gradually extends the protection path until it arrives at the destination node. We defined sets of edges; the sets are formed according to the number of protection paths using an edge. In each step the algorithm makes a local decision.

At first, the available edges are in the set of the most times used edges. If it does not find there an appropriate edge, it goes further through the set of the fewer used edges. If an appropriate edge was found it starts again the search from the set of the most times used edges until it arrives to the destination of the demand.

The algorithm uses back-tracking. If it comes to a dead end, i.e. all sets are checked and there is no appropriate edge, it marks the last used edge as a wrong way, steps back, and tries to find another edge.

What does an appropriate edge mean? The following conditions must be checked: [8]

- it is not used by the working path of the demand
- it does not arrive to a node that is already included in the path, because in this case there would be a loop in the path
- the chosen edge extends the already constructed path
- the protection path on the new edge does not violate the capacity constraint
- the edge is not marked as dead-end

2.2. Algorithm Versions

We analysed two versions of both algorithms. The first version was to route all of the working paths first, and then we protected the working paths ('in blocks' version). The second version was to route the working and the protection path of one demand immediately one after the other for all demands ('in pairs' version).

2.3. Objective Types

We defined two types of objectives. The first was to minimise the number of used edges for protection (MinEdge). This leads to traffic concentration on the selected edges, but yields less administrative work. The second type of objective is to minimise the reserved bandwidth (MinBW), which is related to the operational costs.

By TTD algorithm, the different objectives are carried out in the weighting function of the edges. If it is set up uniformly, the result is the shortest path in hop-count. However, if we set the weights according to the new capacity to be reserved, then the result will be capacity-cost sensitive.

By FS algorithm with *MinEdge* objective an arbitrary edge is chosen, which satisfies the conditions. With *MinBW* objective the edges are sorted according to the new capacity to be reserved on them. The edges are checked in this order against the conditions. Therefore, that edge will be chosen, which requires the fewest new capacity and also fulfils the conditions.

3. Test Environment

We compared the algorithm of the shortest protection path (TTD) and the algorithm of the "as few protection edges as possible" (FS).

3.1. Network Topologies and VPN Input Traffic

We tested the methods on two different network topologies which were generated with the "BRITE" program [9]. We used the "Barabási-Albert" model to generate the topologies. Both topologies had 80 nodes, the difference was only the average edge degree, which was 3.925 and 7.75. The capacity was 1024 unit on each edge.

The traffic was generated by our own program, which generated randomly sized demands between randomly chosen nodes. Two kinds of input traffic was generated, a *homogeneous*, in which all VPNs have 5 members and the bandwidth requirements of the demands are between 45 an 55 units, and a more *inhomogeneous*, in which the VPNs have members between 4 and 8 and the bandwidth requirements are between 30 and 80 units.

3.2. The Simulations

The simulations were run in the following manner. We placed more and more VPNs into the network, until the number of unsuccessfully routed demands reached 1%. On the x axis the number of VPNs is shown.

4. Results

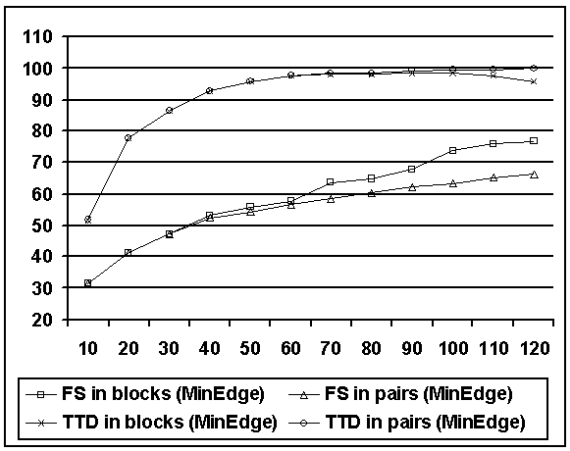
The difference between the homogeneous and inhomogeneous traffic did not show up in the results. Regarding the average node degree the characteristics of the curves was similar. In the following figures we present results with the 7.75 average node degree capacity matrix and with the homogeneous traffic input. The figures show only the protection related metrics, because the results regarding the working paths were the same in all cases.

The FS algorithm uses significantly less edges for protection with the MinEdge objective (Fig. 1(a)). However, with MinBW objective the TTD algorithm is better than the FS (Fig. 1(b)).

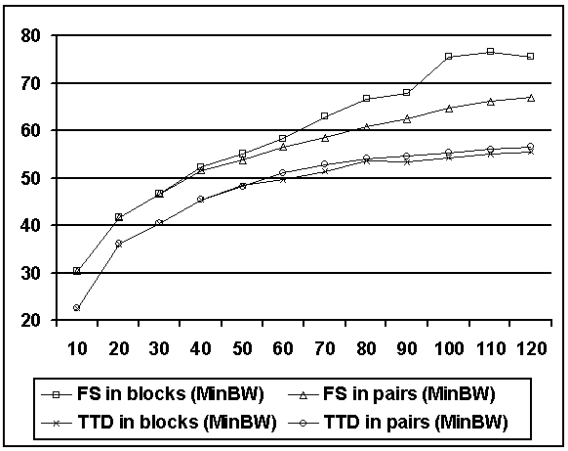
The capacity reserved for protection with MinEdge objective is the following. The FS is worst than the TTD at lower network loads (within 2-3%), but outperforms the TTD over 100 VPNs (Fig. 2(a)). With MinBW objective the TTD performs much better than FS (Fig. 2(b)).

The average protection path length is definitely longer with FS algorithm than with TTD in all objective types and versions (Fig. 3), because FS algorithm concentrates on reusing the edges and not on finding shortest paths.

The 'in blocks' and 'in pairs' versions can also be seen in the figures. The difference at TTD is not significant, but at FS the 'in pairs' is better in the number of protection edges. We examined further the curves, when the unsuccessfully routed demands were over 1%, and the 'in pairs' version had less unsuccessful routed demands than the 'in blocks' version in the same situation.

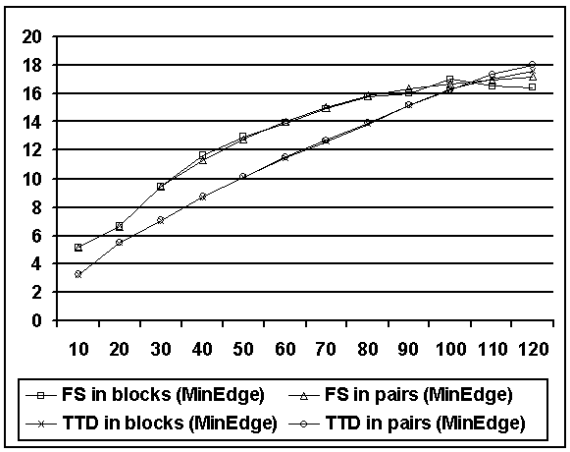


(a) MinEdge Objective Type

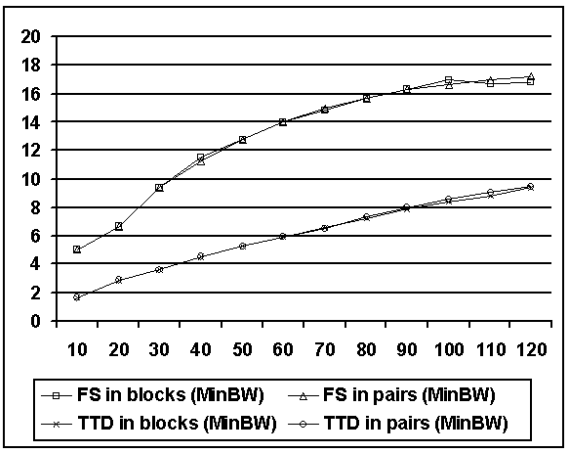


(b) MinBW Objective Type

Figure 1: Number of Edges Used for Protection (%)

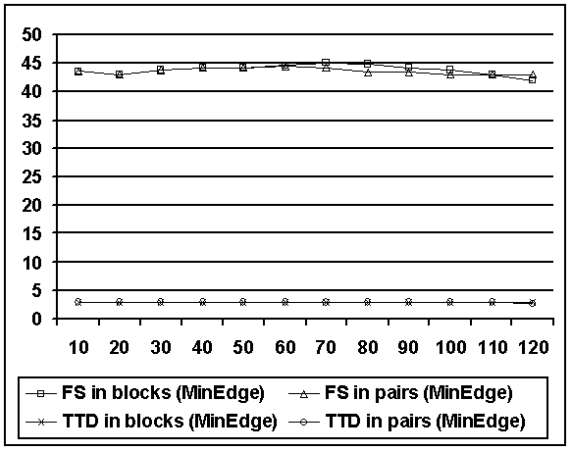


(a) MinEdge Objective Type

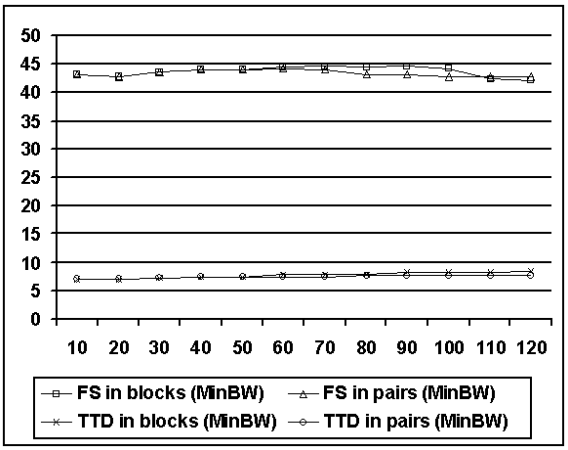


(b) MinBW Objective Type

Figure 2: Number of Edges Used for Protection (%)



(a) MinEdge Objective Type



(b) MinBW Objective Type

Figure 3: Average Protection Path Length (hops)

5. Conclusion

Our results show that there is no significant difference between the ('in blocks' vs. 'in pairs') versions of the algorithms and the homogeneous or inhomogeneous traffic input.

According to the results, if the delay of the connections (which depends on the path length) is the critical factor then Dijkstra's algorithm should be applied (e.g. in VoIP VPNs), because it provides shorter paths. If bandwidth efficiency (MinBW) is the objective (e.g. data VPNs), then also the TTD algorithm yields better results.

However, the FS algorithm performs better when the objective is to minimise the number of used edges (MinEdge). The FS algorithm's novelty is that it does not deal with the shortest path for the demands, it only takes into account the number of demands protected on a link. This advantage shows better when the degree of network connectivity is higher.

References

[1] J. De Clercq and O. Paridaens, "Scalability Implications of Virtual Private Networks", *IEEE Communications Magazine*, May 2002

[2] T. Cinkler and M. Maliosz, "Configuration of Protected Virtual Private Networks", *Design Of Reliable Communication Networks, DRCN 2001*, Budapest, Hungary

[3] P.-H. Ho and H. T. Mouftah, "A Framework for Service-Guaranteed Shared Protection in WDM Mesh Networks", *IEEE Communications Magazine*, Feb. 2002, pp. 97-103.

[4] P.-H. Ho and H. T. Mouftah, "Allocation of Protection Domains in Dynamic WDM Mesh Networks", *10th IEEE International Conference on Network Protocols (ICNP 2002)*, 12-15 November 2002, Paris, France, Proceedings pp. 188-189.

[5] B. Józsa and D. Orincsay, "Shared Backup Path Optimisation in Telecommunication Networks", *Design Of Reliable Communication Networks, DRCN 2001*, Budapest, Hungary

[6] P. Hegyi, M. Maliosz, Á. Ladányi and T. Cinkler, "Heuristic Algorithms for Shared Protection of Virtual Private Networks", *9th Open European Summer School and IFIP Workshop on Next Generation Networks, EUNICE 2003*, Budapest - Balatonfüred, Sept 2003.

[7] S. Yuan and J. P. Jue, "Shared Protection Routing Algorithm for Optical Networks", *Optical Networks Magazine*, vol. 3, no. 3, May/June 2002, pp. 32-39.

[8] V. Subramani, "A Heuristic Routing Algorithm for Shared Protection Optical Network", Project report, CSE990 - Optical Communication Systems/Networks, University of Nebraska - Lincoln

[9] A. Medina, A. Lakhina, I. Matta and J. Byers, Computer Science Department, Boston University, *BRITE: Universal Topology Generation from a User's Perspective*, (User Manual) BU-CS-TR-2001-003. April 05, 2001.