Cluster–based Resource Provisioning for Optical Backbone Networks

Csaba Antal¹, János Harmatos¹, Alpár Jüttner¹, Gábor Tóth¹, Lars Westberg²

¹Traffic Analysis and Network Performance Laboratory, POB 107, H-1300 Budapest, Hungary.
²Ericsson Research, Torshamngatan 23 SE-164 80, Stockholm, Sweden.

This paper presents a novel resource provisioning method for optical backbone networks running IP or MPLS. Trunk and hose models are well known bandwidth provisioning models but both have significant disadvantages if applied to large scale networks. The management complexity of the trunk model highly increases with the size of the network, while bandwidth efficiency of the hose model is often excessively low. We propose an intermediate solution between hose and trunk models. By dividing the network into clusters and using cluster-based traffic description an appropriate equilibrium can be found between management complexity and overprovisioning.

© 2006 Optical Society of America

OCIS codes: 000.1200, 060.0060.

1. Introduction

Offering transport services with flexible Service Level Agreements (SLA) and to achieve high network utilization are important goals of transport network providers. The customer–pipe (trunk) model and the provider–pipe (hose) model are two ways to define bandwidth parameters in current SLAs. The trunk model assumes that the SLA contains maximum traffic demands between each customer site, i.e. there is a traffic matrix with point–to–point traffic demands. Hose model provides a simpler traffic description by using the total incoming/outgoing traffic from the sites in the SLA, i.e. the traffic description includes point–to–anywhere type of parameters.

The point–to–point traffic demands between each site in the trunk model allow the operator to independently reserve bandwidth for these customer pipes. Therefore, the transport provider is able to utilize the network in the best way, since the known traffic matrix determines exactly the required link capacities if routing information is also known. The critical part of this model is that the communication pattern between the end–points is difficult to estimate. Customers may be unable to exactly predict and define traffic loads between the sites, which makes it difficult to specify the complete site–to–site traffic matrix for the Service Level Agreement (SLA). Even if the estimation of the traffic matrix is supported by tools, it is hard to specify the proper bandwidth requirement due to traffic fluctuations. Another drawback of the customer–pipe model is the complexity of the management of trunks. Resource reservations needs to be configured in each source site to each sink site, including policing, shaping and admission control configurations. That is, the number of parameters to be configured is proportional to the square of the
number of sites in the network. Therefore, configuration complexity may become a major drawback of the trunk model in case of large scale networks.

Incoming and the outgoing traffic volumes at each site are needed for the hose model, which can be specified either according to the physical capacity of the link to the provider’s network or based on measurements. Whichever approach is used, the estimation of traffic demand is easier and more precise compared to the customer-pipe model. Here the number of configuration parameters is proportional to the number of sites. These properties make the hose model definitely attractive for customers. The application of the hose model has a great impact on resource provisioning in the optical backbone though. Network dimensioning based on partial information on the traffic demands yields considerable overdimensioning [1] compared to trunk model. Furthermore, the required overprovisioning increases significantly with the size of the network, regarding both the number of sites and the number of links.

We claim that neither of the presented resource provisioning methods scales well to large networks. The trunk model requires the specification and configuration of a large number of traffic parameters, while the hose model requires excessive overprovisioning in the backbone.

We propose a SLA description along with the corresponding dimensioning method that scales to large networks regarding both bandwidth efficiency and configuration complexity. The key idea is to characterize the traffic by point-to-multipoint demands, instead of the point-to-point demands of trunk model and the point-to-everywhere demands of hose model. The new model is called cluster-based model as the notion of clusters of sites is introduced to provide a clear framework for the definition of traffic demands.

It is also the goal of this paper to analyze the proposed cluster-based provisioning model in different network scenarios. Bandwidth efficiency improvement by routing optimization and bandwidth requirement for link and node protection will also be studied and compared to trunk and hose models.

The rest of the paper is organized as follows. After a short overview of related works, Section 3 describes the proposed provisioning method. In Section 4 a linear programming based algorithm for computing the necessary link capacities is given. Section 5 describes the simulation environment, then Section 6 gives a detailed comparison of the different provisioning methods in various situations. In Section 6.B routing methods to further improve on the performance of the proposed method are also described and tested. Finally, results are summarized in Section 7.

2. Related work

Trunk model based telecommunication network optimization has long been present in the literature. In [2] Minoux gives a detailed survey on the possible network design tasks. His method is able to handle different cost models. Pióro et al. [3, 4] developed several methods to solve various telecommunication network design tasks. Other heuristics were proposed in [5] for designing failure protected networks. Other papers deal with the special problems of designing Virtual Private Networks in optical context (see e.g [6]).

However, major obstacles about estimating the real point-to-point distribution of the traffic in a complex network [7, 8, 9] on this topic) led to a flourishing line of research that aims at dimensioning the network directly using only the existing information about the network traffic rather then first estimating the full traffic matrix and dimension the network based on this estimation. A prominent representative of this direction is the hose traffic modeling.
Its main concept has long been present in the literature under the theory of “nonblocking networks” [10]. In [11], the author also examines the hose model based network planning questions using tree routing.

Duffield et al. in [12] were the first to propose this concept for provisioning IP Virtual Private Networks. In their paper an analysis on the bandwidth efficiency of the hose model is presented. This paper inspired research work on developing algorithms for designing minimum cost networks based on hose specifications. Kumar et al. in [13] argued that optimal cost solution for hose realizations shall be based on tree topology, and they proved that the general design problem with asymmetric hoses (different amount of traffic sent and received by the hose) and constrained link capacities is \(\mathcal{NP}\)-hard. The latest important contribution in this field improves the tree–based hose realization by proposing restoration algorithms [14].

A more detailed discussion on the overdimensioning required by the hose model compared to the trunk reservation can be found in [1] by Jüttner et al. The authors investigated the effect of the different bandwidth allocation methods, the network size and the link density on this overprovisioning factor. They also presented a linear programming based method to compute the exact value of the link capacities required by the hose model.

In [15] the authors examine similar question that of [12], but they enable multiplexing of the traffic by using an online demand estimation mechanism and a dynamic admission control based on this estimation.

Erlebach and Ruegg [16] developed an algorithm to find the capacity optimal routing if load sharing is enabled.

3. Cluster-based Provisioning Method

The new provisioning method is proposed for a QoS enabled optical backbone network, with static SLA between the provider and customer. The provider may have a policing function that controls ingress/egress traffic in accordance with the SLA. The customer may have a shaper at network edges for the same purpose. Furthermore, the customer may operate an admission control function for real-time services to avoid QoS degradation due to exceeding the traffic limitations specified in the SLA.

The proposed service model allows to define SLAs – based on the concept of clusters – in a more flexible way than the hose and trunk models. By defining a cluster as a set of sites, intra–cluster provisioning and inter–cluster provisioning can be differentiated in SLAs. Intra–cluster provisioning is concerned with resources between sites of a single cluster, while inter–cluster provisioning refers to resource provisioning between sites that belong to different clusters.

The concept of trunk and hose models can still be used in the context of both intra–cluster and inter–cluster provisioning. In intra–cluster provisioning, trunk model means that SLAs include point–to–point limitations between each site–pair in the cluster. Hose provisioning for intra–cluster traffic means that SLA parameters are traffic demands from sites towards any other sites in the same cluster, i.e. point–to–anywhere traffic demands in the scope of the cluster.

In inter–cluster provisioning, using the notion of trunk and hose models is not so straightforward. It can be applied by considering each external cluster as a single site in the trunk model. That is, trunk provisioning for inter–cluster traffic in that context means that the SLA includes bandwidth parameters for traffic aggregates from each site to each cluster, i.e. site–to–cluster traffic demands are specified. Hose provisioning in the inter–cluster context assumes that the total inter–cluster traffic of the sites is to be specified.
Network provisioning includes both intra–cluster and inter–cluster parts, so four provisioning methods can be defined from the combination of the above methods:

- intra–trunk and inter–trunk
- intra–hose and inter–trunk
- intra–trunk and inter–hose
- intra–hose and inter–hose

The management of this architecture includes tasks for the customer and for the provider too. It is up to the customer to measure traffic in the network and renegotiate SLAs when traffic exceeds a given limit. Configuring the admission control and shaping in accordance with the SLA is also the task of the customer. On the other hand, the provider has to ensure that the bandwidth specified in the SLA is always available in the backbone. In case of renegotiation, the provider has to check if he/she can cope with the increased traffic or some of the links need to be upgraded. It is also up to the provider to configure policing according to the SLAs.

Simple SLA is attractive to customers because it makes the corresponding network management tasks simpler too. On the other hand, under-specified traffic descriptions yield overprovisioning, which makes the offerings more expensive. Therefore, a reasonable balance between management complexity and overprovisioning in the backbone is to be found.

Regarding traffic measurement and SLA renegotiation, the larger aggregates are the subject of the SLA, the easier is for the customer to identify when the SLA is to be renegotiated. In cluster–based provisioning, the required traffic information in the SLA can be adjusted to the traffic information available for the customer.

The form of Service Level Agreements grossly affects the complexity of the configuration of admission control for real-time traffic and shaping for best-effort traffic, i.e. the management complexity. Bandwidth limitations for cluster–based provisioning should be configured for a group of sites, which may be a single site, a cluster or multiple clusters. As shaping and admission control at the customer identify clusters based on the IP prefixes, the configuration entry for a bandwidth limitation in any of these functions consists of a list of IP address prefixes and a bandwidth value. Thus, the configuration complexity not only depends on the number of bandwidth values but on the number of IP address prefixes too. When a cluster is mapped to a single IP prefix then the configuration is clearly less cumbersome than in the case when it consists of a number of disjoint prefixes. Therefore, if clusters are already defined at the beginning then the addressing plan should take them into account.

When considering the responsibilities of the provider, the complexity of policing configuration is similar to that of shaping and admission control at the customer. In addition, the provider has to check if backbone links can cope with the traffic limited by ingress policers. In this respect, the cluster–based method is similar to hose provisioning because complex calculations are needed to fulfill this task. Checking network resources against trunk–based resource requests is much easier as all resource reservation protocols, such as aggregate RSVP or future NSIS protocol for routed IP networks and RSVP-TE for MPLS support trunk reservations.

It is also up to the provider to optimize routing so that the actually used network resources are minimized. Thus, he/she should solve the optimization task with requirements on minimum bandwidth efficiency, maximum number of bandwidth limitations and maximum number of IP prefixes.
4. Capacity Calculation Method

It is shown in this section, how the link capacities can be computed when the cluster based bandwidth reservation method is used. The proposed algorithm is an extension of the method proposed in [1].

In the model, the network is represented by a directed graph $G = (V, E)$ given by the set $V$ of vertices and the set $E$ of edges representing the sites and the links of the network, respectively.

The actual routing between any pair of sites is also assumed to be given. Routing is usually given by a path $p_{uv}$ for each pair of sites $u$ and $v$. However, it is possible to specify the routing in a more general way. For each pair of sites $u$ and $v$ a flow function $r_{uv}: E \rightarrow [0, 1]$ is introduced, where $r_{uv}(e)$ denotes the portion of the traffic between $u$ and $v$ that goes on the link $e$. In this way, both the single path routing (by setting $r_{uv}(e)$ to 1 on the edges of the path between $u$ and $v$ and to 0 on the other edges) and the shared routing can be handled.

Once the routing is given, a given traffic matrix determines the load on the links. If $t_{uv}$ denotes the amount of the traffic from $u$ to $v$ then the traffic of a certain link $e$ is

$$tr(e) := \sum_{u,v \in V} r_{uv}(e) t_{uv}. \hspace{1cm} (1)$$

The network dimensioning method presented in this paper does not assume the knowledge of the actual traffic matrix, but only some side constraints on the traffic matrix and aims at designing a network that is able to carry any traffic that meets the given side constraints.

4.A. Preconditions

The side constraints to be given express what is known about the traffic in advance or what can be measured. The side constraints can be classified as follows.

**Trunk Parameter.** When specifying this type of side constraint, the maximum amount of the traffic from a certain given site $u$ to another one $v$ is assumed to be known. The maximal value is denoted with $T_{u \rightarrow v}$. This constraint can be formalized as follows.

$$t_{uv} \leq T_{u \rightarrow v} \hspace{1cm} (2)$$

**Hose Parameter.** In case of the traditional hose traffic description the limits for the traffic originated from and directed to a certain site $u$ are specified by $T_{u \rightarrow V}$ and $T_{V \rightarrow u}$, respectively. In mathematical formulae it means

$$\sum_{v \in V} t_{uv} \leq T_{u \rightarrow V} \hspace{1cm} \text{and} \hspace{1cm} \sum_{v \in V} t_{vu} \leq T_{V \rightarrow u}. \hspace{1cm} (3)$$

**Cluster Based Parameter.** This constraint specify limits for the traffic between a site $u$ and an arbitrary set $S$ of sites (e.g. a cluster). Similarly to the previous case, two values used for describing the traffic, one for the outgoing and another one for the incoming traffic, $T_{u \rightarrow S}$ and $T_{S \rightarrow u}$, respectively. In mathematical formulae

$$\sum_{v \in S} t_{uv} \leq T_{u \rightarrow S} \hspace{1cm} \text{and} \hspace{1cm} \sum_{v \in S} t_{vu} \leq T_{S \rightarrow u}. \hspace{1cm} (4)$$

Note that all the above constraints are linear making it possible to solve efficiently the optimization problems that use these constraints.
The capacity reservation method presented in this paper is based on a clustering of the sites. Thus, the set of the sites is partitioned into \( k \) disjoint subsets called clusters, i.e. \( V = C_1 \cup C_2 \cup \cdots \cup C_k \), and \( C_i \cap C_j = \emptyset \) whenever \( i \neq j \).

The idea is to differentiate between the intra-cluster and the inter-cluster traffic. In both cases there are two natural possibilities.

4.A.1. Intra-cluster traffic

Consider the cluster \( C_k \), and the site \( u \) sitting in \( C_k \). Then the intra-cluster traffic of this site can be given in the following two ways.

**Intra-trunk.** In this case, the exact amount of traffic of \( u \) to each sites in \( C_k \) is given, by using the parameter \( T_{u \rightarrow v} \) for each \( v \in C_k, u \neq v \).

**Intra-hose.** In this case, only the limits for the sum of the outgoing (and incoming) traffic of \( u \) goes to (or comes from) another site in the same cluster is given, by using the parameter \( T_{u \rightarrow C_k} \) and \( T_{C_k \rightarrow u} \).

4.A.2. Inter-cluster traffic

Consider again the site \( u \) sitting in \( C_k \). Similarly to the above cases, there are two ways for describing the traffic of \( u \) outside its cluster \( C_k \), as follows.

**Inter-trunk.** In this case the exact amount of traffic of \( u \) from and to each cluster is given, by using the parameters \( T_{u \rightarrow C_j} \) and \( T_{C_j \rightarrow u} \) for each \( j \neq k \). Note, that this is a weaker description than the traditional point-to-point trunks.

**Inter-hose.** In this case only the limits for the sum of the outgoing (and incoming) traffic goes to (or comes from) another site in another cluster are specified for each site, by using the parameter \( T_{u \rightarrow V \setminus C_k} \) and \( T_{V \setminus C_k \rightarrow u} \).

4.B. Capacity calculation

As it was mentioned, the aim is to dimension the network so that it is able to carry any possible traffic demand that meets the preconditions. Thus each individual link must be dimensioned considering the worst-case scenario. To compute this maximum traffic of a link \( e \), the traffic matrix \( t_{uv} \) that meets the preconditions and maximizes the traffic value (1) must be found.

As the objective function (1) and the constrains (2), (3), and (4) are linear functions, the value of (1) can be maximized efficiently by any linear programming method. (In the numerical evaluations the simple lp.solve software package [17] was used). Of course, in order to get the total necessary bandwidth, this process has to be repeated for each edge \( e \in E \).

**Note,** that \( r_{uv}(e) \) is zero for most of the \( u,v \) pairs, so the size of the real linear program to solve can be largely reduced by omitting each variable \( t_{uv} \) whose corresponding route does not use the edge \( e \). The constraints that does not affect any remaining variables can also be omitted.

As an example, let us see the linear program where the intra-trunk/inter-hose traffic description is used.

\[
\max \sum_{u,v \in V} r_{uv}(e)t_{uv} \quad (5a)
\]
subject to

\[ t_{uv} \in \mathbb{R} \quad \forall u, v \in V, u \neq v \]  
(5b)

\[ t_{uv} \geq 0 \quad \forall u, v \in V, u \neq v \]  
(5c)

\[ t_{uv} \leq T_{u \rightarrow v} \quad \forall k, \forall u, v \in C_k, u \neq v \]  
(5d)

\[ \sum_{v \in V \setminus C_k} t_{uv} \leq T_{u \rightarrow V \setminus C_k} \quad \forall k, \forall u \in C_k \]  
(5e)

\[ \sum_{v \in V \setminus C_k} t_{vu} \leq T_{V \setminus C_k \rightarrow u} \quad \forall k, \forall u \in C_k \]  
(5f)

5. Simulation Environment

To demonstrate the advantages of the proposed architecture, simulations were carried out to test the new provisioning method on a realistic topology with realistic traffic distributions. Topologies of large IP backbone networks are publicly available, which made it easy to define the example network. The network that was used in the performance analysis of this paper is based on the AT&T backbone network [18] shown in Figure 1.

For the purpose of the present study, the site-size statistics presented in [19] were used to estimate the generated load in sites of the investigated network. During the dimensioning process only voice sources were assumed to be present in the system, which generated calls according to Poisson arrival process. Thus, each site in the network was assigned an offered load value, assuming a fairly big network. The offered load of the largest site was 4500E, the smallest one 100E and the total offered load in the network was 35400E. After having the generated calls in the sites, the offered load matrix was calculated by distributing the originated calls proportionally to the size of destination sites in terms of generated calls (i.e. the more calls are generated in a site, the more it receives from the others).

The input bandwidth parameters for the dimensioning process were determined by applying the Erlang B dimensioning formula for the traffic aggregate that shares a common bandwidth limit in the SLA and in the admission control. When calculating the bandwidth parameters, a target blocking probability of 0.1% was assumed.

As it was described before, the dimensioning methods work on clustered networks. It can be easily seen that the structure of the clusters affects the required capacity, even if the number of the clusters is kept the same and only the arrangement of the sites to the clusters is varied. To partition the network sites into clusters a heuristic algorithm was used. The clusters formed by the algorithm are guaranteed to be connected sub-networks in themselves. The other objective of the algorithm was to balance the size of the clusters in terms of sites. Finding the optimal arrangement at a given number of clusters is out of scope of this paper.

6. Performance Study

In this section the performance evaluation of the proposed cluster based provisioning method is presented. Firstly, the four proposed variants are compared without considering routing optimization and protection methods. Then it is studied how the routing optimization can be used to improve the bandwidth efficiency. Finally, the effects of protection methods on the results are investigated.
6.A. Comparison of cluster–based provisioning variants

The comparison of the provisioning variants is based on shortest path routing and assumes that no protection methods are applied. The required capacity of the links and the management complexity are the key measures characterizing the performance of the methods, therefore they were calculated for the evaluation. The studied network scenario is based on the 25 node AT&T network (Fig. 1) and the pre-calculated traffic matrix. Dimensioning was performed for each possible number of clusters (from 1 to 25). The cluster–based provisioning variants are introduced in Section 3, from now on the following short names are used to refer them:

- 'tt': Trunk provisioning for intra–domain and inter–domain traffic
- 'th': Trunk provisioning for intra–domain traffic and hose for inter–domain traffic
- 'ht': Hose provisioning for intra–cluster traffic and trunk for inter–cluster traffic
- 'hh': Hose provisioning for intra–cluster and inter–cluster traffic

![Fig. 1. Topology of the studied network](image1)

![Fig. 2. Overprovisioning factor with different dimensioning methods](image2)

Figure 2 shows the overprovisioning factor which is defined as the relative difference between the capacity need of the evaluated method and the trunk model. Note, that although the pure hose and trunk provisioning is not displayed explicitly, their results can be seen in the figures, as they are equivalent to specific cases of the cluster–based methods: If there is one cluster then method 'ht' and 'hh' is equivalent to hose model and methods 'tt' and 'th' correspond to trunk model. The results for the cluster based methods are between the hose and trunk model results except for some points below the x axis. These points indicate that the cluster based methods can over-perform the trunk provisioning by exploiting the Erlang gain of multiplexing voice trunks.

Figure 3 presents the average number of bandwidth limitations per site, which is closely related to management complexity. It can be seen that values for the four cluster based methods are between that of hose and trunk models. The figures also highlight the a trade-off between bandwidth efficiency and management complexity.
Fig. 3. Management complexity with different dimensioning methods

Fig. 4. Management complexity as the function of overprovisioning at different dimensioning methods

To compare the variants, the real question is the necessary management complexity using a clustering that provides a certain targeted overprovisioning. The management complexity – overprovisioning scatterplot in Figure 4 compares the variants in this point view. The dots represent the management complexity and overprovisioning values for the above cluster configurations. It can be seen that method 'ht' is the best in this sense on the AT&T network: it requires the least management complexity for any fixed overprovisioning. For example, by allowing 20% extra bandwidth in the backbone over the requirement of the trunk model, the needed configuration parameters in a site decreases from 25 to 5. Note that hose model would require 130% extra bandwidth with a single parameter in each site.

The reason why hose provisioning is better than the trunk method for intra–cluster provisioning is that link capacities inside the cluster are not very sensitive to the provisioning method, but trunk model needs much more configuration parameters than hose. The small difference in efficiency is because the topology inside a cluster is typically close to a tree, which is the optimal scenario for the hose model [1], due to the sparse topology of the example AT&T backbone network.

In the rest of the paper, the focus is on the performance of this method from different aspects.

6.B. Routing methods

Previous studies have shown (see e.g. [20, 2, 5] for traditional trunk based network design and [1] for hose model) that the choice of routing may have significant effect on bandwidth efficiency, both in trunk and hose dimensioning. For pure trunk dimensioning obviously the best choice is when the traffic is routed via the least hop path, while tree routing (i.e. when the traffic is routed via a spanning tree) gives the best performance for the hose dimensioning.

In the cluster–based provisioning method, clustering divides the network into two levels, which motivates the investigation of the effect of multi-level routing solutions. Applying shortest and tree routing on both level 4 additional routing scenarios can be defined as follows.

- 'ss': Shortest path intra-cluster / Shortest path inter-cluster
- 'st': Shortest path intra-cluster / Tree inter-cluster
- 'ts': Tree intra-cluster / Shortest path inter-cluster
In the case of multi-level routing, the cluster-level mechanism has precedence over the site level algorithm, which means the following: If shortest path routing is used both on the intra-cluster and inter-cluster level, then routing paths are chosen in such a way that they cross as few inter-cluster links as possible, and among these paths the least hop path is selected. This principle is applied to the other three routing scenarios as well.

Figure 5 shows the overprovisioning factor of method 'ht' at the five investigated routing (Simple shortest plus the four routing strategies described above). The basic conclusion that can be drawn based on Figure 5 is that using tree routing on the inter-cluster level results worse performance than shortest path routing. The reason for the bad performance of the inter-cluster tree is that it disables direct connection between many clusters, resulting in large detours. One can also observe that using tree or shortest path routing on the intra-cluster level does not make a significant difference. This is because the analyzed network is relatively sparse, thus the routing paths in the tree and shortest routing case are very similar. In case of more dense networks the tree routing on the intra-cluster level should perform better than the shortest path routing.

Similar tests were made for the other cluster based provisioning methods as well, and the results confirmed the expectation that using tree routing where hose dimensioning is applied and shortest path routing in network segments dimensioned based on trunk model is the best choice regarding the routing. Thus, the best routing type for each of the cluster-based methods can be selected as follows.

- routing 'tt' for method 'hh'
- routing 'ts' for method 'ht'
- routing 'st' for method 'th'
- routing 'ss' for method 'tt'.

The performances of the four methods with their best routing were also investigated. For the comparison, Figure 6 shows again the overprovisioning factor – management complexity scatterplot for each method.
The situation is very similar to that of using shortest path routing for each, but two differences can be observed. One is that one can not state that method 'ht' is the overall best, because if the overprovisioning factor is higher than 40%, some 'th' and 'hh' configurations requires less management complexity to achieve the same overprovisioning, though the difference is almost negligible. The other difference is that many points belonging to the 'ht' and 'tt' curves are located at the right hand side of the y axis. This means that there are many options to achieve better performance than the trunk model with less management complexity.

6.C. Protection Methods

In backbone networks one of the most important requirement is fault tolerance. This fact motivated the following tests to examine how much overprovisioning is needed to provide certain fault tolerance using traditional dimensioning methods and the proposed cluster based schemes.

Tunneling techniques used in optical backbones can be different. The major difference regarding protection methods is if the backbone is a routed IP network or an MPLS-based network.

When the backbone is a routed IP network then the route of packets is determined based on the actual content of the routing tables. As a result of a link failure, the routing tables of the affected routers will be updated by routing protocols. When all routing tables are updated based on the changed link state information, the packets are routed via the shortest path considering only the remaining links. This process may take a few minutes. It also means that the protection path of a given flow depends on the failed link.

If the backbone is an MPLS network, then the advanced failure handling features of MPLS can be used. One of the techniques for protection in MPLS is using backup label switched paths. That is, two LSPs are set up between each pair of sites, a primary and a secondary. When all links are up then the primary LSP is used for communication. Whenever a link along the path of the primary LSP fails, traffic is rerouted to the secondary LSP. To ensure that the secondary LSP can be used in case of any failure along the path of the primary LSP, the two LSPs must be disjoint. As LSPs are set up before the actual link failure, the protection path of a given flow is independent of which link is failed. An advantage of this technique is the much faster fail-over time than IP routing.

6.C.1. Capacity calculation

The effects of protection methods were examined both in routed IP networks and in MPLS with backup LSPs. During the investigation shared protection was assumed for single link and single node failures. That is, the network dimensioning was performed in such a way that links will support the rerouted traffic in case of any link or any node fails, but only one at a time. When a node fails, all of its links are removed from the topology and its traffic is also removed from the traffic matrix.

6.C.2. Native IP Backbone

All the variants of the new provisioning method, their dependency on the proposed routing methods were studied in the same way as the previous cases when no protection was considered. The tests indicated that applying protection has not significant influence on the relative performance of the four variants of the cluster-based provisioning method. Therefore, only the protected counterpart of Figure 6 is demonstrated.
Figure 7 shows the corresponding management complexity—overprovisioning factor pairs for the optimized routing strategies. In this case the overprovisioning factor is defined as the relative difference between the given method and the trunk model using shortest path routing and applying protection. As it was mentioned, the results show similar characteristics to the protection-less case, except for the fact that there are more cluster-based configurations that require significantly less network capacity than trunk model does here. It is because applying this kind of protection method, the shortest path routing is not the optimal choice for trunk model. Note that the optimal routing strategy for trunk model can only be obtained using explicit route definitions and the calculation of these routes is a complex optimization problem. It can be observed that the curve of the ’ht’ method is almost vertical at the left side where the points represent the cases where the network was split into many clusters. Therefore, in the examined AT&T network scenario, splitting the network into more than 5 clusters – left to point (-6%,5) on the curve – makes no sense because the achievable additional overprovisioning gain over that point is negligible.

6.C.3. MPLS Based Backbone

The effect of a path-protecting redundancy mechanism on the performance of cluster–based provisioning methods was also investigated. First two disjoint paths were determined using Edmond’s minimal cost flow algorithm [21]. Among them the one with lower cost were chosen to be primary and the other one to be the secondary path.

The same tests were performed as for the protection mechanisms of native IP backbone. The overprovisioning factor – management complexity graph is plotted for this protection case in Figure 8. The displayed results are similar to that of the other protection method, but in this case there is no such cluster–based configuration which could over-perform the trunk provisioning. Another difference is that the curve of the ’ht’ method is not as steep as in the other case. The reason for this is that the applied routing strategies are developed to force the primary route to the aimed path in a routed IP network. In contrast, path protection is based on Edmonds’ algorithm, which chooses the primary and secondary paths such that the summed costs of the two paths be minimal, so the shortest path and the primary
path could be different. For example when there is no disjoint alternative path to the shortest one, but there are two other disjoint paths in the network, then Edmonds’ algorithm will choose those ones. Thus splitting the network into 5 clusters seems to be the best choice here for the ‘ht’ method.

7. Conclusion

The main contribution of the paper is the cluster–based provisioning method, which by dividing the network into clusters makes it possible to define point–to–multipoint(cluster) SLAs between the providers and customers. The method is a generalization of the well-known hose and trunk models combining the advantages of both in the case of large networks, where neither methods scale due to low bandwidth efficiency and large management complexity, respectively. The behavior of the proposed method depends on the number of clusters. The two extrema represent the traditional hose and trunk methods.

A linear programming based algorithm was also proposed for congestion free network dimensioning using the above traffic model. The congestion-free network design allows the customer to use non-adaptive real-time services in optical networks, which would be degraded in case of congestion.

Performance evaluation was carried out to compare the variants of the method and to study the effect of routing optimization and protection methods.

Route optimization was shown to decrease the overdimensioning of the cluster–based method, in the above example of five clusters from 20% to 5%. A two-level routing strategy was the best for the selected intra–cluster–hose inter–cluster–trunk model, which applied tree routing inside the cluster and shortest path routing between clusters.

Tolerance for single failures in a routed IP network required 50% extra capacity for the trunk model. By using the cluster–based method with five clusters, the total link capacity increased with another 20% (with 70% compared to trunk without protection). The gain of routing optimization, however, almost disappeared when protection methods were also applied. Simple shortest path routing, which is independent of cluster definitions was the best routing strategy.

References and Links


