Universität Stuttgart

INSTITUT FÜR
KOMMUNIKATIONSNETZE
UND RECHNERSYSTEME
Prof. Dr.-Ing. Andreas Kirstädter

Copyright Notice

© 2020 VDE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the VDE.

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-68026, Fax: ++49-711-685-67983
Email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

Delay-Differentiated Routing in Meshed Backbone Networks

Tobias Enderle, Arthur Witt, and Filippos Christou
Institute of Communication Networks and Computer Engineering, University of Stuttgart, Stuttgart, Germany
{tobias.enderle, arthur.witt, filippos.christou}@ikr.uni-stuttgart.de

Abstract

Today's network applications require strict quality of service (QoS) guarantees, e. g., in terms of availability or delay. Due to shortest path routing in backbone networks, the point-to-point delay is usually very low. However, this is not only true for the services with strict delay requirements but also for all others. Since there is no customer-perceivable delay difference, it is difficult for the network operator to justify differences in pricing between low-delay and other services. We propose a routing and network reconfiguration approach for packet-optical networks that enforces delay differentiation by routing on non-shortest paths. We present an integer linear program formulation and study the effects of our approach by simulation in two backbone topologies. We show that a considerable number of services can be routed with delay differentiation if the use of additional spectral resources is allowed.

1 Introduction

During the last years, many new types of network applications have seen the light of day, e. g., high quality video streaming, smart manufacturing or connected vehicles to name a few. Those new types of applications put much more diverse requirements on the underlying networks than it was the case in the past. Some require large data rates, others have very strict quality of service (QoS) requirements, like very high availability or low jitter and delay. The latter in particular has even led to new network architectures comprising edge data centers, which shows how important QoS is for today's applications.

In backbone networks QoS compliance is usually achieved by overprovisioning of network capacity [1, Ch. 8]. Having available a lot of spare capacity helps to avoid congestion [2] and provides protection capacity for the event of failures in the network. As a result, the delay can be kept at a minimum, bounded only by processing, serialization and the physically implied propagation delay. The availability requirements can be supported by adding protection.

Apart from overprovisioning, other mechanisms have been developed to provide QoS compliance, most notably IP's differentiated services (DiffServ). DiffServ can help maintain QoS when network resources are scarce by prioritizing selected packets. However, since most network service providers (NSPs) prefer to overprovision their networks, occasions in which such additional mechanisms are beneficial appear rarely.

In summary, one can assert that providing a sufficient level of QoS is no real problem in backbone networks as long as the network utilization is low enough. All traffic can be handled equally well. This makes selling different service classes with increasingly strict QoS parameters difficult because for the customer there are no perceivable differences between those classes when the network is in a normal state [1, Ch. 8]. Even if a service class with better QoS values is offered, customers might misuse classes with worse guarantees, knowing that no perceivable differences exist.

In the following we focus on one particular QoS parameter, namely the point-to-point delay that services in the network experience. Services are usually routed along a physically short path because it results in low delay and good availability. For services with strict delay requirements, providing the lowest possible delay is imperative, while for other services higher delays are tolerable. However, since shortest path routing is generally applied to all services, no differentiation in delay exists. Therefore, a price difference is hard to justify and, consequently, there is no way for the NSP to capitalize on services with strict delay requirements.

A similar motivation has been reported for the transatlantic submarine cable "Hibernia Express". Since 2015 it provides the fastest fiber optic connection between New York and London. The low-delay connections are sold at high prices. However, since not all customers require such low delay, more affordable connections are offered as well. To justify the price difference, artificial delay is introduced by additional fiber lines. [3]

QoS differentiation has been studied from many different angles, e. g., from the protocol perspective with DiffServ, as mentioned before. Other works on availability [4] or setup delay tolerance [5] focus on traffic engineering aspects. However, to the best of our knowledge, none of the existing approaches enforce service differentiation to justify different price levels.

In this work we provide a way to introduce perceivable delay differentiation in meshed backbone networks. We propose an approach that routes services with different delay requirements on paths of different lengths. In particular, we place low-delay services on the path with the lowest delay, while other services are routed along paths with a higher delay. In that way, we enforce the differentiation between service classes. This provides a possible justification for price differences and makes misuse of service

classes difficult. We have presented a first step towards this approach in [6], where we reduce the delay overfulfillment, i. e., the difference between required and actual delay of a service, by routing on non-shortest paths. This time, instead of only routing on paths different from the shortest one, we route traffic on paths that guarantee a perceivable difference in delay. Additionally, our approach does not require more router ports than our hardware-optimized reference, which guarantees cost efficient network operation.
In the next section we will first introduce the problem statement in detail. Afterwards we present the concept of our proposed approach for service differentiation. In Section 3 we formally introduce our solution approach including an integer linear program (ILP) formulation and auxiliary algorithms. Section 4 presents an illustrative numerical example in which we study the behavior and effects of our approach in a US and a German wide-area network. Finally, we conclude the paper in Section 5.

2 Problem Statement

2.1 Motivation

We consider that an NSP, owner of a backbone network, offers at least two delay-distinguishable service classes which are defined in a service level agreement (SLA). An Internet service provider (ISP), a customer of the NSP, buys connectivity under the SLA conditions from the NSP. In such networks shortest path routing is often used due to its simplicity and good performance regarding the computation effort, the resulting propagation delay and resource usage.
Routing traffic on the shortest paths is a good strategy, especially for services that require low latency. Considering a typical network, we can observe that the shortest path delays between individual node pairs vary a lot due to the different geographical distances between them. Often, NSPs guarantees a maximal average delay for their network in an SLA [7]. Typically, this value is higher than the average delay that is measured for all node pairs in the network. Naturally, the shortest path delay for distant node pairs will be higher than the guaranteed average delay mentioned above. Therefore, instead of the network-wide average, we have to consider the delay of the shortest path between individual network nodes as a reference if we want to discuss the service requirements, service quality or routing methods for low-latency services. E. g., if the shortest path delay exceeds the required delay of a low-latency service, it will be impossible to provide this service for the corresponding node pair. In Figure 1 path delays per node pair are shown exemplary.
The considered traffic to route in the network is grouped according to the service's delay requirement and a service classification, as depicted in Figure 2a) and c) respectively. Figure 2a) shows that a portion of the traffic is of delay-sensitive nature, which may contain traffic from augmented reality applications or machine-to-machine communications. The larger traffic portion is delay-insensitive, i. e., it does not have any delay requirements. Further, we assume that this traffic grouping based on the required latency is only known by the ISP and not by the NSP.

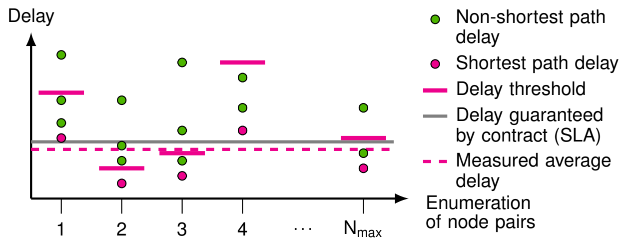


Figure 1 Path delays based on propagation delays for individual node pairs.

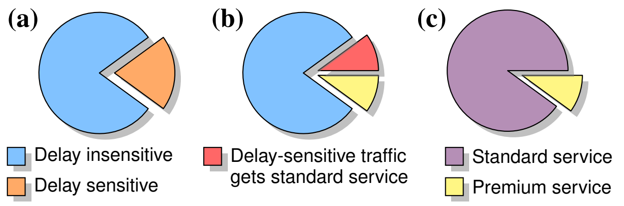


Figure 2 Traffic shares grouped by **(a)** delay requirement and **(c)** service class. In **(b)** the delay-sensitive traffic is further separated by an ISP: As shortest path routing can be assumed, a portion of the delay-sensitive traffic is treated as standard network service to save money.

The second traffic grouping, see Figure 2c), is based on the service classes offered by the NSP: The premium network service class will transport the traffic with delays as low as possible, i. e., by applying shortest path routing. The standard service class does not comprise any delay requirements itself. However, due to economical reasons, shortest path routing or hardware optimized routing, which also leads to short paths and low delays, is often used.
As a consequence, the ISP can use the standard service even for (a portion of) the delay-sensitive traffic, see Figure 2b). The achieved delay will be close to that of the premium class with a high probability. So, the ISP can save money due to the lower cost of the standard service, compared to the premium service, without service degradation in terms of the provided delay.
It is very difficult for the NSP to set up its services such that he can urge the ISP to use the services correctly. Using longer paths for the standard class traffic to increase the delay would introduce a clear service differentiation in delay, but the following problems would arise:

1. Redirecting the huge amount of standard service traffic on longer paths will increase the amount of required resources.
2. Delay-sensitive and delay-insensitive traffic are mixed in the standard class traffic. Distinguishing them for the purpose of redirecting only the delay-sensitive portion is nearly impossible.

2.2 Solution Approach

We investigated in an approach, named as differentiated service delay (DSD) routing, that is able to increase the delay for the traffic of the standard service class in a way that the delay for a portion of the delay-sensitive traffic in this service class is significantly increased compared to the premium class. The aimed consequence is that due to this introduced delay-based service differentiation, the ISP has to use always the premium class for the transport of the delay-sensitive traffic to avoid delay-requirement violations. We have also to consider that the network resources are a valuable asset and should be saved.

This target can be reached with a procedure that applies the following strategy:

1. For each node pair a delay threshold $\tau_{s,t}$ (s: source node, t: target node) is defined, that is related to the delay for the shortest path, e. g., two times the shortest path delay, see Figure 1. Successful service differentiation is applied if a portion of the standard class traffic (in the best case the delay-sensitive part of it) can be routed with a delay higher than this threshold, see Figure 5.
2. The premium traffic is routed on the shortest paths that are available. Therewith, the lowest possible delays with respect to the propagation delay can be reached.
3. The traffic of the standard class is treated with the three following operations, see Figure 3:
   (a) Traffic demands, that should be routed with an increased delay, are preselected to reduce the amount of traffic that should be routed on longer paths, and also the final amount of additional activated network hardware. For the reason of fairness, the preselection is done randomly in predefined time intervals.
   (b) Each preselected traffic demand is split into portions with a fixed maximum traffic amount.
   (c) A subset of the preselected traffic portions is routed on longer paths to increase the delay beyond the predefined delay thresholds $\tau_{s,t}$. The remaining traffic is routed arbitrarily in terms of delay.

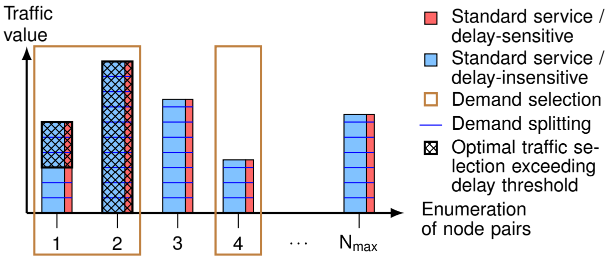


Figure 3 Operations on the standard class traffic.

The proposed routing approach overcomes the problems mentioned in Section 2.1. Further, it has a few degrees of freedom, which can be used as design parameters as long as the principles mentioned above are used. First, the traffic portion size of the demand splitting can be chosen. For a small size, the approach can adjust the traffic amount to the shorter or longer paths with a finer granularity. On the contrary, it has to solve a more complex problem. Second, the demand selection allows to vary the amount of demands that are considered for a routing on longer paths.

To calculate how many times on average we have to repeat the demand selection until we pick all the available $N_{\max}$ end-to-end demands of the network at least once, we adapted the equation of Stadje [8] to our scenario. In Figure 4 we evaluate this equation for several network sizes of

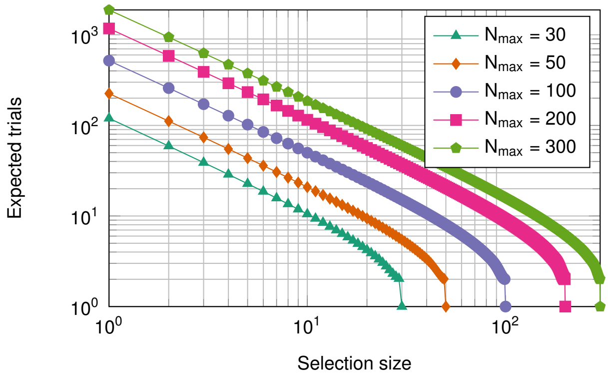


Figure 4 Expected number of trials needed to select at least once all the possible end-to-end demands ($N_{\max}$) of a network.

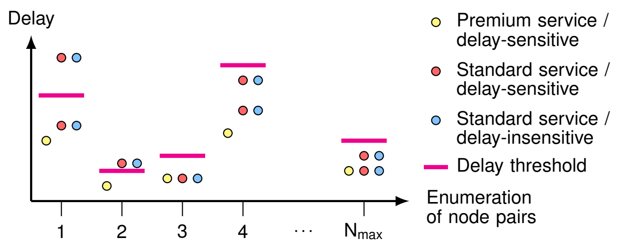


Figure 5 The proposed approach creates the lowest delay for the premium service, and larger delays for the different traffic portions of the standard service. Due to traffic splitting, multiple delay values per traffic kind may occur.

different $N_{\max}$, depending on the selection size. We see that the expected trials are growing exponentially as the selection size is reduced. In addition, for all cases the expected trials must be 1 when the selection size equals $N_{\max}$, which explains the knee of the curves at $N_{\max} - 1$.

The expected delay behavior of the DSD routing is shown in Figure 5. The delay values are given for traffic portions that are on the same path, i. e., split traffic demands can be routed on more than one path which ends up in one or more delay values. While the premium service is routed on shortest paths (cf. Figure 1), the standard traffic is routed with a tendency to longer paths and higher delay. We expect a delay differentiation which is large enough to separate the delay-sensitive from the delay-insensitive services.

3 Optimization

In this section we present an ILP formulation that realizes the solution approach introduced in Section 2.2. The formulation is based on our previous work in [6].

We consider a meshed transport network consisting of a circuit-oriented optical layer and a packet-oriented electrical layer on top. The fiber topology is a directed graph $G(V,E)$ with nodes V and fiber links E. Each node consists of a color-, direction- and contentionless optical cross connect and a packet router connected to it. We assume that the cross connect is reconfigurable in an software-defined networking (SDN) fashion. Further, we assume the router ports to be tunable, such that they terminate an optical circuit as it is the case in IP-over-DWDM architectures. The principle idea

Algorithm 1 Generation of candidate paths.

Require:

P set of all simple paths in G from s to t

k maximum number of candidate paths to return

ℓ function to determine the delay of a path $p \in P$

```
function FindCandidatePaths(P, k, ℓ)
    L ← {argmin_{p∈P} ℓ(p)}                         ▷ lowest delay path
    while (|L| < k) ∧ (L ≠ P) do
        L ← L ∪ {argmax_{p∈P\L}(min_{l∈L} |ℓ(p) − ℓ(l)|)}
    return L
```

can be applied to any packet-optical architecture though.

We reconfigure the network, i. e., the optical circuits and the traffic routing, regularly to handle changing traffic demands. For the migration between two configurations we employ a "make before break" approach, i. e., connections in the old configuration can only be removed after the replacing connection is set up. In each reconfiguration step we have a set of (split) traffic demands D for which connectivity must be provisioned. For each demand $d \in D$, h_d describes the requested data rate. We also have the set $D_S \subseteq D$ which contains demands of the standard class that have been selected for routing on threshold-exceeding paths.

We employ an ILP to find optimized network configurations. The ILP is based on a link-path formulation, i. e., the optimizer selects traffic routes from a set of candidate paths, which we generate in advance.

The optimization target in our proposed approach is the routing of the demands in D_S on non-shortest paths in order to cause perceivable service differentiation between those demands and those of the premium class.

The details of the path generation and the ILP are presented in the following sections.

3.1 Generation of Candidate Paths

Our ILP realizes a link-path formulation which depends on preselected candidate paths. A standard approach to the generation of candidate paths is the use of a k shortest path algorithm, where the number of paths, k, can be used to adjust the complexity of the ILP. Since most routing problems focus on short paths, this is a valid approach. However, in our case we are interested in longer paths as well in order to differentiate services. Therefore, we generate candidate paths according to Algorithm 1.

In order to find the candidate paths from node s to node t, we first compute all corresponding simple paths. We then select k of those simple paths as candidate paths in an iterative manner. We first select the path with the lowest delay. Then we successively add paths which have the largest difference in delay from the already selected ones. In that way, the candidate paths represent a variety of path delays.

Since we consider large-scale, high-performance transport networks, we do not expect any major queuing delays in the network nodes [2]. Instead, the main contribution of delay is propagation delay in the fiber links. Therefore, we model the delay of a path by the propagation delays of the links it traverses. Our ILP constrains the planned circuit utilizations to stay below an adjustable value η. In that way, the probability of congestion and the resulting queuing delays is reduced greatly.

A candidate path describes the end-to-end route through the network. However, since we allow grooming of multiple demands into a single circuit, a demand is not necessarily routed on a direct optical circuit in the optimal configuration. Instead, it might traverse a sequence of circuits, such that nodes along the candidate path are either bypassed or used for optical-electrical-optical conversion and packet routing. Hence, for each demand $d \in D$, the set L_d contains the candidate paths and the set $R_{d,l}$, with $l \in L_d$, contains the corresponding circuit path sequences. For each candidate path, $\varepsilon_{d,l} \in \{0,1\}$ equals 1 if the path delay is below the delay threshold $\tau_{s,t}$ of the corresponding node pair. For demands in the premium class we only generate one candidate path to ensure shortest path routing.

3.2 Integer Linear Program Formulation

The basic task of the ILP model is to determine a candidate path and a corresponding sequence of circuit paths for each demand. Based on this selection, it computes the required amount of active optical circuits and ports to carry all demands. In doing so, the ILP must adhere to resource limitations like the number of available ports at each node or the capacity of the links.

We define the set U that contains all node pairs, i. e., $U = \{(u,v) \in V \times V : u \neq v\}$. The set C consists of all circuit paths that result from the various $R_{d,l}$, i. e., each $c \in C$ is a potential circuit path on which a number of actual optical circuits can be activated. The ILP then contains the following variables:

- $g_{d,l,r} \in \{0,1\}$ $(d \in D, l \in L_d, r \in R_{d,l})$: path selector, i. e., $g_{d,l,r}$ equals 1 if demand d is routed over path l using the circuit path sequence r
- $w_c \in \mathbb{N}$ $(c \in C)$: the number of active, parallel circuits on circuit path c
- $w'_c \in \mathbb{N}$ $(c \in C)$: the number of active, parallel circuits on c taking migration into account
- $p_{v,t} \in \mathbb{N}$ $((v,t) \in U)$: the number of ports at v for connections with t
- $p'_{v,t} \in \mathbb{N}$ $((v,t) \in U)$: the number of ports at v for connections with t taking migration into account
- $i_v \in \{0,1\}$ $(v \in V)$: indicator for a highly utilized node, i. e., i_v equals 1 if a certain amount of the installed ports is occupied at node v

Furthermore, the ILP contains the following constants:

- $\xi \in \mathbb{R}$: the data rate of a single optical circuit
- $\pi \in \mathbb{N}$: the maximum allowed number of optical circuits on a link
- $\psi \in [0,1]$: the soft utilization limit of nodes; if the fraction of occupied ports is higher, the indicator i_v is activated
- $\eta \in [0,1]$: the maximum allowed utilization of an optical circuit
- $\rho_{c,d,l,r} \in \{0,1\}$: indicates whether circuit path sequence r of demand d's candidate path l uses circuit path c

- $\delta_{e,c} \in \{0,1\}$: indicates whether circuit path c traverses link e
- $\varphi_{c,u,v} \in \{0,1\}$: indicates whether circuit path c connects nodes u and v with u as source node
- $\widehat{g}_{d,l,r} \in \{0,1\}$: the path selector of the previous configuration (a constant for the current optimization)

We can now define the constraints as follows (the set $\widehat{D}$ contains the demands that are already present in the network):

$$\sum_{l\in L_d, r\in R_{d,l}} g_{d,l,r} = 1 \qquad \forall d \in D \quad (1)$$

$$\sum_{d\in D, l\in L_d, r\in R_{d,l}} \rho_{c,d,l,r} \cdot h_d \cdot g_{d,l,r} \le \xi \cdot w_c \qquad \forall c \in C \quad (2)$$

$$\begin{aligned}&\sum_{d\in D, l\in L_d, r\in R_{d,l}} \rho_{c,d,l,r} \cdot h_d \cdot g_{d,l,r} + \sum_{d\in \widehat{D}, l\in L_d, r\in R_{d,l}} \rho_{c,d,l,r} \cdot h_d \cdot \widehat{g}_{d,l,r} \\ &- \sum_{d\in D\cap\widehat{D}, l\in L_d, r\in R_{d,l}} \rho_{c,d,l,r} \cdot h_d \cdot g_{d,l,r} \cdot \widehat{g}_{d,l,r} \le \eta \cdot \xi \cdot w'_c \qquad \forall c \in C\end{aligned} \quad (3)$$

$$\sum_{c\in C} \delta_{e,c} \cdot w'_c \le \pi \qquad \forall e \in E \quad (4)$$

$$\sum_{c\in C} w_c \cdot \varphi_{c,v,t} \le p_{v,t} \qquad \forall (v,t) \in U \quad (5)$$

$$\sum_{c\in C} w_c \cdot \varphi_{c,t,v} \le p_{v,t} \qquad \forall (v,t) \in U \quad (6)$$

$$\sum_{t\in V\setminus v} p_{v,t} \le \chi_v \qquad \forall v \in V \quad (7)$$

$$\frac{1}{\chi_v} \sum_{t\in V\setminus v} p_{v,t} \le \psi + i_v \qquad \forall v \in V \quad (8)$$

$$\sum_{c\in C} w'_c \cdot \varphi_{c,v,t} \le p'_{v,t} \qquad \forall (v,t) \in U \quad (9)$$

$$\sum_{c\in C} w'_c \cdot \varphi_{c,t,v} \le p'_{v,t} \qquad \forall (v,t) \in U \quad (10)$$

$$\sum_{t\in V\setminus v} p'_{v,t} \le \chi_v \qquad \forall v \in V \quad (11)$$

Equation (1) ensures that a demand is routed on exactly one path. Constraints (2) and (3) ensure that enough circuits are activated depending on the chosen paths. In contrast to constraint (2), which only considers the demands of the current reconfiguration step, constraint (3) also takes the demands of the previous configuration into account allowing a "make before break" migration. Equation (4) is a link capacity constraint. Constraints (5) and (6) activate a sufficient amount of router ports to accommodate the active optical circuits. Constraint (7) limits the number of active ports at node v to the amount of installed ports χ_v. Constraint (8) activates the indicator i_v if the fraction of occupied ports at a node v exceeds the predefined value ψ. In that way, headroom for later migrations is preserved. Constraints (9) – (11) have the same purpose as (5) – (7), but also consider migration. Finally, the optimization target for our proposed network reconfiguration ILP is

$$\begin{aligned}\min\Bigg(&\frac{\alpha}{|V|}\sum_{v\in V} i_v + \beta \sum_{(v,t)\in U} p_{v,t} + \gamma \sum_{c\in C}(w_c + w'_c) + \\ &\frac{\delta}{\sum_{d\in D_S} h_d} \sum_{d\in D_S, l\in L_d, r\in R_{d,l}} \varepsilon_{d,l} \cdot g_{d,l,r} \cdot h_d\Bigg).\end{aligned} \quad (12)$$

Notice that the last term only considers the preselected demands in D_S. For the cost factors we choose $\alpha \gg \beta > \gamma > \delta$. In that way, migration headroom is the primary goal of the optimizer. Whenever headroom capacity is not a problem, the number of active ports and optical circuits is the main minimization target leading to resource efficient network configurations. The last term is what makes our approach special. It penalizes demands in D_S which are not routed on threshold-exceeding paths.

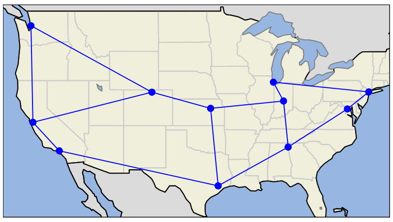
4 Illustrative Numerical Example

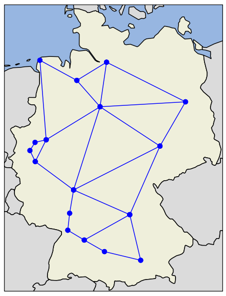
4.1 Scenario

For the evaluation of our approach we have used a US topology and a German topology based on the Abilene and the Nobel backbone networks openly available in SNDlib [9]. The Abilene topology was slightly modified by removing a node playing the role of a leaf vertex, and more particularly only connected to Atlanta. Such special nodes are very rare to find in a meshed backbone network and we considered them unrelated to our scenario. The two network topologies are illustrated in Figure 6.

In both cases we have assumed an IP-over-DWDM architecture using up to $\pi = 80$ channels and a single line rate setup with router ports transmitting at a rate of $\xi = 100$ Gbps under a transparent optical reach of 2500 km. To aid migration, the ILP will discourage configurations in which more than 80 % of the ports installed at a node are occupied ($\psi = 0.8$). For each of the two networks, we set the maximum number of candidate paths that Algorithm 1 can choose and the number of maximum circuit sequences per candidate path, in order to reduce the computation effort. While for the German topology a maximum of 5 candidate paths and 5 circuit sequences are allowed, in the US network we take into account all paths as candidate paths and all of their circuit sequences because of the limited linkage complexity of the network. Additionally, we permitted configurations with $\eta = 0.7$ meaning up to 70 % utilization for each circuit. This way our reconfiguration method avoids high link utilization in the network which also enables us to disregard queueing delay in our model [2]. Furthermore, the delay threshold value $\tau_{s,t}$ between end-to-end nodes is computed as two times the delay induced from the corresponding shortest path.

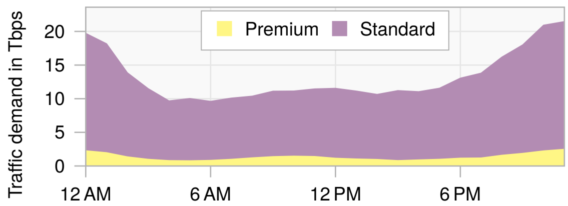
The network traffic was artificially generated through the procedure described in [10]. The total demand produced for the US network is given in Figure 7a). The demand for the German network is of similar characteristics and volume. Each demand connects a node pair with a required data rate and can either be delay-sensitive or delay-insensitive, based on the delay requirements of the underlying application. We assume that the delay-sensitive traffic is 20 % of the whole traffic and that the NSP is unable to distinguish between them. In this sense, the NSP can only distinguish demands based on the SLA describing them. As mentioned, the existence of an SLA with two service classes, standard and premium, is examined. The premium traffic only consists of the 50 % of the delay-sensitive traffic, which means the 10 % of the whole traffic. Premium demands are always



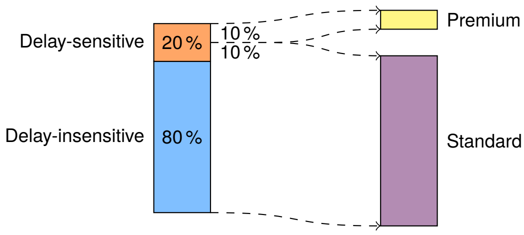


(a) US topology (avg. link length: 993 km) **(b)** German topology (avg. link length: 143 km)

Figure 6 The network topologies.



(a) Total traffic demand for the US topology



(b) Demand outline

Figure 7 The traffic model.

routed on the shortest path. The remaining 50 % of the delay-sensitive traffic (10 % of total) is handled with the standard service class together with the delay-insensitive traffic. Demands belonging to the standard class are handled with the DSD methodology to provide differentiated routing. Figure 7b) displays the above described notion, similarly to Figure 2. These percentages apply, with some randomness included [10], not only for the aggregate network traffic but also for every single end-to-end demand.

4.2 Reference Approaches

We compare the proposed approach with two reference approaches to highlight the effects. The first reference approach is a shortest path routing (SP). It is realized by the ILP presented in Section 3.2 with a modified cost function in which the last three terms are replaced by one term that penalizes the physical path length. In that way, all demands are routed on the shortest path in terms of length and consequently also propagation delay. The second reference is a hardware-oriented optimization (HW) which is realized by the ILP as well. This time, only the last term, which penalizes demands below the delay threshold, is removed from the cost function. Consequently, the resulting configurations require a minimal amount of router ports.

4.3 Simulation

For the implementation of our DSD methodology, the high-performance mathematical programming solver IBM CPLEX was used. Due to the time-consuming search for the global optima of the solver, it was decided a time limit of 40 minutes for every optimization. It was observed that with this time limit the ILP optimizations of DSD and HW ended with an average gap of 1.6 % while the SP reached 0 %. To balance the randomness introduced from the time limit, we conducted 30 differently seeded optimization runs per simulated point in time at a granularity of one hour. Running the optimization multiple times with different initialization for the solver is still important for other reasons. For a single set of demands different optimal network configurations are possible and the solver can only select one of them. Therefore, the network state may evolve differently over time for separate runs. Besides that, the seeded simulations help to regularize the results drawn from the randomized process of demand selection.

Before running the evaluative simulation, it is still left to determine the number of installed ports χ_v per node. For this reason we first run a dimensioning simulation to determine the number of ports needed. This simulation adheres to the HW optimization described in Section 4.2 and uses the same traffic as described in Section 4.1. The results of the dimensioning were overprovisioned with a factor of 1.25 yielding the values χ_v of Section 3.2.

We tested our approach for a series of demand selection sizes. The demand selection changes every 6 hours and each time we randomly pick another subset of D. As a result, this yields 22 expected trials for the German network, given that each time we select 68 end-to-end demands out of the overall 272 (the 17 nodes of the German network makes $17 \cdot 16$ end-to-end demands). This means that in the specific scenario of the 6 hour rotation of D_S, the algorithm will have gone through all the demands at an expected time of 5 days and a half.

4.4 Results

Figures 8 and 9 show the results for the introduced scenarios for the course of a day. We show values for each of the three routing approaches. In all plots DSD is represented by three graphs, one for each selection size. Each of these graphs is the average of 30 optimization runs. In Plots a), c), d) and e) the SP and HW approaches are represented by a single graph (the average over 30 optimization runs and the three selection sizes) because the selection size is not relevant for them. Plot b) shows the percentage of selected traffic that is routed on a threshold-exceeding path. Since this quantity depends on the selection size, also SP and HW are represented by three graphs. Premium demand is not included in the plots.

Figure 8 shows the results for the US topology. In Plot a) it is visible that SP requires a considerable amount of additional ports compared to the other approaches. This is an expected behavior because the other approaches are designed to minimize active ports. The port usage of DSD is only slightly higher (3.2 % on average) than that of HW. Hence, there is no disadvantage in using DSD.

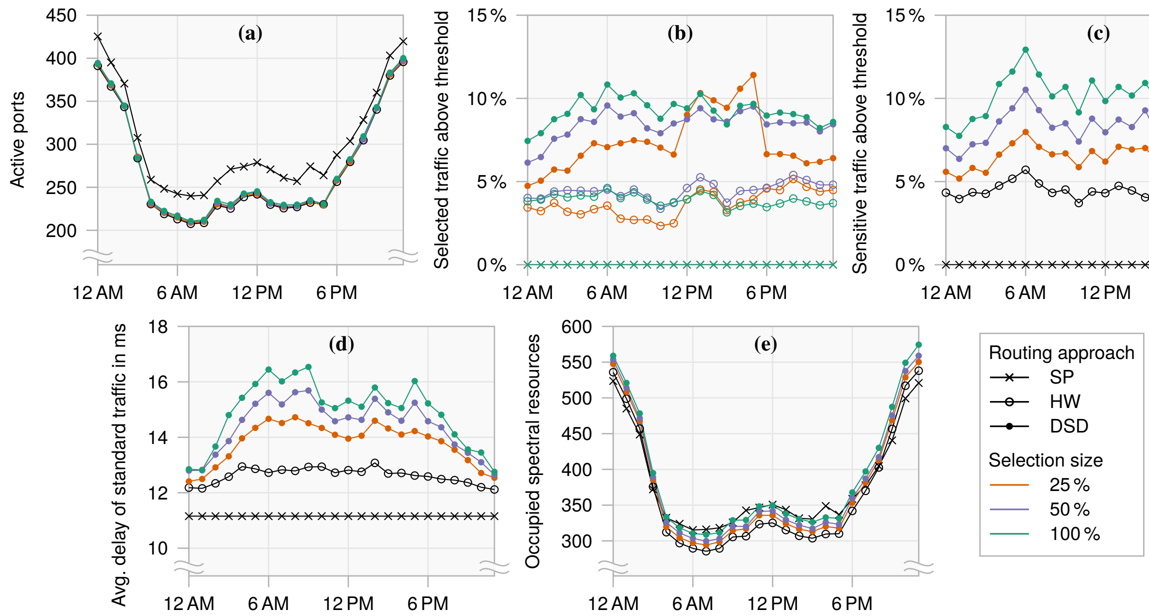


Figure 8 Simulation results for the US topology.

Figure 8b) considers the selected traffic, i. e., the traffic created by the demands in D_S, and shows the percentage that is routed on threshold-exceeding paths. This percentage experiences a propagation delay at least two times as high as the corresponding premium demands. As expected, the percentage is zero for SP because everything is routed on the shortest path. For DSD the percentage varies between 4.7 % and 11.4 %. While the values for a selection size of 50 % and 100 % are quite stable, the curve for a selection size of 25 % shows a considerable shift from 12 PM to 6 PM. This illustrates the sensitivity to changes in the demand selection if only a small number of demands is selected. The curves also reveal that under high traffic loads, which is the case at the beginning and the end of the day, less traffic can be routed above the threshold due to resource constraints. Interestingly, the HW approach exhibits a considerable percentage of threshold-exceeding traffic as well. HW treats every demand equally. Nevertheless, since it employs grooming to minimize active ports, some demands are routed on threshold-exceeding paths "by accident". In particular, there is no way to control which demands exceed their delay threshold using HW.

The selected traffic consists of delay-sensitive and insensitive traffic. Since our main goal is to create a delay differentiation for the sensitive traffic, Figure 8c) shows the percentage of sensitive traffic that is routed on a threshold-exceeding path. It can be seen that the percentage increases with the selection size. There are two reasons for this. First, with an increased selection size the amount of selected sensitive traffic also grows. Second, with an increased selection size the diversity of selected demands grows and it is easier for the optimizer to find demands that are suitable for routing on long paths. However, an increased selection size also means less control about which demands experience delay differentiation.

The fact that a considerable number of demands is routed on paths that exceed their delay threshold also influences the network-wide average delay of the standard service class. In Figure 8d) it is visible that the shortest path routing of SP results in a constant average delay of about 11 ms. Using DSD the delay increases to almost 13 ms during high network utilization (around 12 AM) and rises up to 16.5 ms for a selection size of 100 %.

The last plot, Figure 8e), explores the amount of occupied spectral resources. The amount of spectral resources a single circuit occupies equals the number of links in E the circuit traverses. The plot shows the summation for all active circuits. As can be seen, DSD occupies more resources than HW. For a selection size of 100 % the difference is 25 resource units on average. This shows how DSD achieves long paths, namely by increasing the lengths of individual optical circuits. For low network loads (from 5 AM to 5 PM) DSD occupies less spectral resources than SP, but for higher loads it surpasses both SP and HW.

We have evaluated our DSD approach in the German topology as well. The results are presented in Figure 9. Compared to the US topology, the German network is of much smaller geographical extent. However, since we use the same assumptions for the ports, in particular a transparent reach of 2500 km, the maximum relative length of an optical circuit wrt. the network extent is much higher. As a result, much higher percentages of traffic can be placed on threshold-exceeding paths. As can be seen in Figure 9b), for a selection size of 100 % more than 60 % of the selected traffic can intermittently be routed on paths that exceed the delay threshold. This means that more than half of the total traffic of the standard class is affected. For the sensitive traffic in Plot c), between 37 % and 57 % are achievable. While these values are much higher than those observed in the US topology, also the differences induced by the different selection sizes are larger. Furthermore, in Figure 9e) it is visible that the large amounts of traffic that experience delay differentiation also require a considerable amount of additional spectral resources. The average difference between DSD

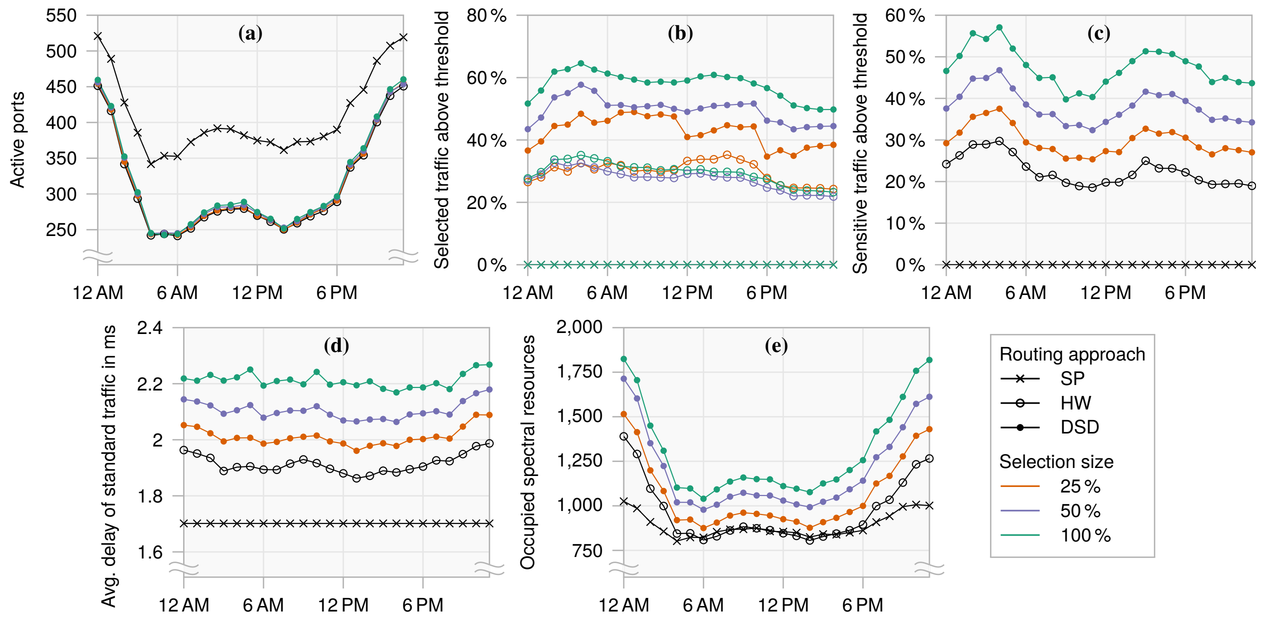


Figure 9 Simulation results for the German topology.

and HW for a selection size of 100 % is 340 spectral units, while compared to SP it is 416 spectral units. However, for high network loads around 12 AM the resource usage for HW is already much higher than for SP. This suggests that the goal of minimizing active ports alone is an important cause for increased usage of spectral resources.

5 Conclusion

In this paper we have introduced a routing and network reconfiguration approach that enforces customer-perceivable delay differences for network traffic of different service classes. This is achieved by routing traffic of the premium service class on paths with the lowest delay while traffic of other classes is routed on paths that exceed a certain delay threshold. Using the ILP formulation for our proposed DSD approach, we have shown for two backbone networks that a considerable amount of delay-sensitive traffic can be routed under differentiation without increasing the amount of active router ports. In the considered scenarios this portion of traffic experiences at least two times the delay of the premium class. It is also visible that the differentiation consumes more spectral resources than a hardware-optimized or shortest path routing. However, by adjusting the selection size for DSD, the NSP is able to control the trade-off between the amount of differentiated traffic and the additional consumption of spectral resources.
The achievable delay differentiation can be used to justify pricing differences between the service classes. This provides a way for the NSP to capitalize on low-delay services.

6 Literature

[1] XiPeng Xiao. *Technical, Commercial and Regulatory Challenges of QoS: An Internet Service Model Perspective*. Morgan Kaufmann, 2008.

[2] Baek-Young Choi et al. Analysis of point-to-point packet delay in an operational network. *Computer Networks*, 51(13):3812–3827, 2007.

[3] Tim Stronge. 10 Weird (and Possibly Useful) Things You Didn't Know about International Networks. *NANOG65, Montreal, Canada*, October 2015. [Online]. Available: `https://www.youtube.com/watch?v=uy46gEst72k` (last accessed: 14.09.2020).

[4] Abdulaziz Alashaikh, David Tipper, and Teresa Gomes. Exploring the logical layer to support differentiated resilience classes in multilayer networks. *Annals of Telecommunications*, 73(1-2):63–79, 2018.

[5] Ajmal Muhammad et al. Effect of Delay Tolerance in WDM Networks with Differentiated Services. In *2011 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference*, pages 1–3. IEEE, 2011.

[6] Uwe Bauknecht, Tobias Enderle, and Arthur Witt. Reduction of Delay Overfulfillment in IP-over-DWDM Transport Networks. In *23rd Conference On Optical Network Design And Modelling (ONDM 2019)*, Athens, Greece, May 2019.

[7] NTT. Our Global IP Network SLA. [Online]. Available: `https://www.gin.ntt.net/support-center/service-level-agreements-slas/our-global-ip-network/` (last accessed: 14.09.2020).

[8] Wolfgang Stadje. The Collector's Problem with Group Drawings. *Advances in Applied Probability*, 22(4): 866–882, 1990.

[9] Sebastian Orlowski et al. SNDlib 1.0–Survivable Network Design Library. In *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, Spa, Belgium, April 2007.

[10] Uwe Bauknecht and Tobias Enderle. Modeling Dynamic Traffic Demand Behavior in Telecommunication Networks. In *ITG-Symposium in Photonic Networks 2018*, Leipzig, Germany, June 2018.