

An Optimization-Heuristic Approach to Dynamic Optical Bypassing

Frank Feller
Institute of Communication Networks and Computer Engineering (IKR)
Universität Stuttgart, Pfaffenwaldring 47, 70569 Stuttgart, Germany
Email: frank.feller@ikr.uni-stuttgart.de

Abstract

Due to the growing traffic volume, energy consumption is becoming a major concern in transport networks. Dynamic optical bypassing is one mechanism to reduce both energy consumption and resource requirements. In this paper, we propose a dynamic optical bypassing algorithm based on simulated annealing. This algorithm is execution-time optimized in view of periodic network reconfiguration e. g. every 15 minutes. An adjustable reconfiguration penalty assures stable network operation by limiting circuit set-ups and tear-downs. Evaluating the algorithm in a 15-node reference network, we obtain good bypass configurations in less than one minute. Moderate reconfiguration penalties efficiently reduce circuit modifications without affecting energy efficiency.

1 Introduction

While the traffic volume in communication networks grows exponentially, network operators face an increasing cost pressure. One important cost component is the energy consumption of network equipment. Traditionally, the energy consumption of core networks has been considered negligible compared to other parts of the network. With the sprawl of optical access technologies (FTTx) providing higher access speeds without increasing the energy consumption of the access network, the situation is changing [1]. Hence, mechanisms to reduce the energy consumption in the core are highly desirable.

In addition to the general growth trend, the traffic volume varies on different time scales. The dominant characteristic is a diurnal profile following human activity [2]. These variations, with typically only 25 % of the peak load at night, offer a great potential to save energy by switching off unused network resources.

The emergence of mega data centers hosting cloud services and applications is likely to change traffic characteristics at a scale relevant to core networks. The data centers will account for a significant share of traffic. Cloud applications may migrate between data centers, which entails an according change of the traffic load in the network. A flexible use of network node resources for different transmission paths allows to serve the dynamic traffic with less hardware than required in case of static network operation, thus allowing CAPEX savings.

Today's transport networks are multi-layer networks. They generally consist of an optical lower layer, which is circuit-switched at the granularity of wavelengths or fibers, and electrical upper layers, of which the topmost is often packet-switched. A typical configuration is IP/MPLS in the upper layer and wavelength switched optical networks (WSO) in the lower. Since optical switching is signif-

icantly more energy-efficient than electrical (packet) processing, it is advantageous to switch traffic primarily in the optical layer – provided that the optical circuit is reasonably utilized. Dynamic optical bypassing describes the dynamic set-up and tear-down of optical circuits. It allows reconfiguring a multi-layer network for an energy and resource efficient operation under varying traffic load.

The reconfiguration of network resources cannot happen in arbitrarily short time. This is particularly true for the set-up of optical circuits, which is said to take in the order of 15 to 30 minutes assuming currently deployed technology (though official numbers are not available). A second reason for limiting the frequency as well as the extent of network reconfigurations is the stability of network operation. The high reliability of transport networks is attributed to the traditionally static and thus simple operation mode. While cost pressure and energy efficiency targets will require a change of this paradigm, approaches limiting operational dynamics are likely preferred by network operators.

The limited speed and frequency of network reconfigurations imply that faster traffic fluctuations cannot be exploited for energy savings. In addition, a prediction of the traffic load for the respective reaction time is required. The current technology allows dynamic optical bypassing only to follow diurnal fluctuations (and other predictable patterns). On the upside, less frequent reconfigurations allow the use of centralized approaches to determine the next network configuration. At the expense of an increased signaling and processing load, centralized mechanisms usually find more efficient configurations than distributed approaches.

In this paper, we present a centralized dynamic optical bypassing algorithm based on the optimization heuristic of simulated annealing. This algorithm is designed in view of on-line application, i. e. it finds a good solution in the

order of minutes rather than a (near-)optimal one in much more time. Parameters allow adjusting the trade-off between optical circuits and electrical switching. In addition, a configurable penalty for setting up or tearing down circuits constrains the number of such switching operations. We discuss the effect of these parameters on different metrics by applying the algorithm to a reference network.

The remainder of this paper is structured as follows. Section 2 shortly addresses related work, while section 3 introduces the background on optimization. In section 4, we detail our algorithm. Section 5 presents the scenario and results of a first evaluation of the bypassing algorithm. We conclude in section 6.

2 Related Work

Finding optimal dynamic optical bypass settings essentially means solving multi-layer network optimization problems for changing traffic load situations.

2.1 Multi-Layer Network Optimization

Multi-layer network optimization and traffic grooming have been studied extensively for both static and dynamic traffic [3, 4]. Such problems comprise the following questions:

- finding the optimal virtual topology, i. e. the topology of the lower layer including bypasses,
- routing the traffic in this virtual topology (i. e. the routing of the upper layer),
- routing the bypasses into the physical topology (potentially under resource constraints), and
- assigning wavelengths to the connections of the virtual topology (potentially under continuity constraints).

One frequent approach is to integrate several or all of these questions into the formulation of one optimization problem. While this is the only way to determine the optimum, it leads to highly complex problems, which are only practically solvable for small networks. One counter-measure is to deal with the questions in an isolated, sequential way. In addition, researchers can abstract from some of these questions.

Frequently, exact solutions of the optimization problem are determined by integer linear program solvers. Alternatively, the authors of [5] use an optimization meta-heuristic, genetic algorithm. Simulated annealing, another meta-heuristic, is applied to a related problem in [6].

2.2 Dynamic Network Reconfiguration

There is also an extensive body of literature on the dynamic reconfiguration of networks. Authors mostly assume a global view of the network. For instance, the issue has been considered for hybrid networks (e. g. optical migration capable networks with service guarantees (OpMiGua) [7]). Frequently, the dynamic network configuration prob-

lem is again solved exactly by optimization techniques. The authors of [8] use this approach to evaluate the benefit of allowing different degrees of flexibility in the two network layers. Alternatively, heuristics are applied, e. g. in [9]. In addition, distributed approaches which work on a local view of the network topology and traffic load (e. g. [10, 11]) have attracted some interest.

3 Combinatorial Optimization

Finding an optimal network configuration is a combinatorial optimization problem. This class of optimization problems is characterized by an enumerable, finite solution space. One can formulate such problems as integer linear programs (ILP). Since most of these problems have a high complexity (they are mostly NP-hard), only small problem instances allow for an exact solution in a reasonable time. For problems of practical size, we have to resort to heuristic approaches.

3.1 Optimization Meta-Heuristics

Optimization meta-heuristics describe procedures to search the solution space of complex optimization problems in a randomized way in order to find near-optimal solutions. In general terms, they derive new candidate solutions (neighbor solutions) from a set of current solutions and determine their cost. The next set of solutions is selected from all these solutions depending on the cost. The best solution encountered during this procedure is retained.

Decisive components of an optimization heuristic are thus the so-called perturbation strategy to derive new candidate solutions and the selection strategy. The perturbation as well as the cost computation is essentially problem-specific. A meta-heuristic can only provide coarse guidelines how to derive neighbor solutions (e. g. from one solution or by combining several ones). It is generally more specific on the cost-dependent selection strategy. The combination of perturbation and selection shall assure both diversification and intensification. The former means covering distant points in the solution space in order to escape local optima, whereas the latter designates the search for the optimum within a restricted neighborhood. More details on heuristic optimization and several meta-heuristic algorithms are found e. g. in [12].

3.2 Simulated Annealing

Simulated Annealing [13] is a meta-heuristic inspired by the annealing process in solid-state physics. It iteratively modifies one candidate solution. Solutions of lower cost are always accepted as new candidate solution, whereas more expensive solutions are accepted with a probability that decreases with the temperature of the annealing process. This shifts the weight from diversification to intensification in the course of the process. Figure 1 illustrates this iterative search in the multi-dimensional solution space.

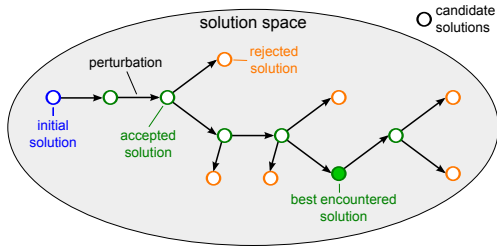


Figure 1 Illustration: iterative search of the solution space

Simulated annealing is known to find good results in relatively short time [12, 14].

The simulated annealing procedure is governed by the cooling schedule. It defines the initial temperature, the temperature length (number of iterations per temperature step), the cooling ratio (reduction of the temperature value per step), and the frozen state, i. e. the termination condition. All these parameters need to be tuned for the respective optimization problem.

The current temperature T defines the probability P_A to accept a solution of higher cost. In accordance with the physical analogon, the probability is obtained by

$$P_A = e^{-\frac{C_{\text{acc}} - C_{\text{cand}}}{T}} \quad (1)$$

where C_{acc} is the cost of the currently accepted solution (the starting point of the last perturbation) and C_{cand} is the cost of the newly derived solution. The initial temperature T_{init} shall provide for an initial acceptance ratio of 90 % to 99 % (including lower-cost candidate solutions). The temperature reduction upon each annealing step may for instance be linear or geometric, i. e.

$$T_i = T_{i-1} - \Delta T \quad \text{or} \quad (2)$$

$$T_i = T_{i-1} \cdot \alpha \quad (3)$$

where α is less than but close to 1. The latter has the advantage that the number of cooling steps does not need to be pre-defined.

Regarding the remaining parameters, the simplest approach is a fixed temperature length and a termination after a fixed number of cooling steps. Alternatively, the termination can be conditioned on the recent success in finding improved solutions. Section 4.3 gives more details on the simulated annealing procedure in its application to the dynamic bypassing problem.

4 Dynamic Bypassing Algorithm

We propose a dynamic optical bypassing algorithm based on simulated annealing. It is intended to be used for periodic adaptation of the network configuration to changing traffic demands. In accordance with technological limitations of the setup speed of optical circuits, we envisage reconfiguration intervals in the order of 15 to 30 minutes. Hence, one major design goal is the applicability in

this scenario, i. e. convergence to reasonably good network configurations within minutes. For this reason, we restrict the optimization to the bypass setting, while the traffic demands are routed deterministically along fixed geographical paths (on which they use available bypasses).

In the following sections, we discuss some assumptions on resources and network nodes, specify the optimization problem, and detail the optimization procedure.

4.1 Model Assumptions

The current version of our algorithm assumes unlimited hardware resources (optical transponders and packet-switching capacity) in all nodes. This allows evaluating different parameterizations of the algorithm with respect to the required hardware resources by simulation. The hardware resource overhead required to set up new circuits before tearing down previous ones (*make before break*) is currently disregarded. For application in practice, the algorithm will need to consider resource constraints. In the simplest case, this is possible by disallowing (i. e. skipping) solutions which do not match the constraints.

Our primary optimization criterion is the energy consumption of the respective network configuration. We describe the energy consumption by the number of optical circuits and the amount of transit traffic in the upper network layer. We thus account for the resource assignment in the granularity of circuits in the optical layer. While we currently assume a fixed cost per optical circuit, the algorithm is easily extendable with a circuit-length dependent cost component accounting for amplification and signal regeneration. In the electrical layer, we assume that the energy consumption scales linearly with the load. While this is arguably not the case for current network equipment, frequency scaling mechanisms and sleep modes implemented in state-of-the-art general purpose processors suggest that future energy-optimized network nodes will show such a behavior. While minimizing the energy consumption, the algorithm shall limit the number of switching operations (set-ups and tear-downs of optical circuits) in order to assure stable network operation. We therefore include a reconfiguration penalty into the cost function.

With these assumptions, we may reduce the general multi-layer optimization problem to the first two items of section 2.1: the definition of the virtual topology (i. e. the bypass setting) and the routing of traffic demands (which is done deterministically). We abstract from the questions of bypass routing and wavelength assignment, which are pointless in case of infinite resources.

4.2 Optimization Problem Description

In order to apply a meta-heuristic, we do not need a mathematical formulation of the optimization problem. Problem instances and candidate solutions are rather represented by software objects comprising the information detailed in the

following.

4.2.1 Problem Instance

An instance of the bypass optimization problem consists of the basic network topology, the demand matrix, and the previous setting of active circuits:

- The basic topology is the physical topology of the lower network layer. It is defined as a graph with directed, weighted edges, where the edge weight represents the cost for routing traffic along this edge. Note that this cost is not considered for the optimization but only to solve the initial routing problem.
- The demand matrix defines the (directed) traffic rates between each pair of nodes. These rates give the aggregate of all connections and traffic flows entering and leaving the network at the respective nodes.
- The previous network configuration specifies the number of (directed) optical circuits existing between each pair of nodes prior to the current reconfiguration step. These circuits either span single hops in the basic topology, or they represent bypasses. This information is required to evaluate the reconfiguration penalty.

4.2.2 Candidate Solution

A solution to the bypass optimization problem is essentially defined by a set of directed edges representing the bypasses. Further data is not required since traffic routing is done deterministically. We do however retain some additional information established during cost computation in order to facilitate the evaluation of the reconfiguration penalty based on a previous solution.

4.3 Optimization Procedure

This section details the central simulated-annealing based optimization algorithm along with the major problem-specific components.

Figure 2 gives an abstract view of the optimization algorithm. First, the current temperature T and the accepted solution S_{acc} along with its cost C_{acc} are initialized. The same applies to the variables for the best encountered solution S_{best} and C_{best} , respectively. Section 4.3.2 describes the cost computation. As initial solution S_{init} , we select the previous bypass setting. This approach is justified by the objective of limiting reconfigurations. Nevertheless, applying this initial solution might require switching operations in case the traffic demand has changed since the last reconfiguration.

In the main loop of the algorithm, a candidate solution S_{cand} is derived from S_{acc} (cf. section 4.3.1) and its cost C_{cand} is computed. Then we first adjust the counter N_{noImpr} of consecutive iterations without improvement of C_{acc} . If

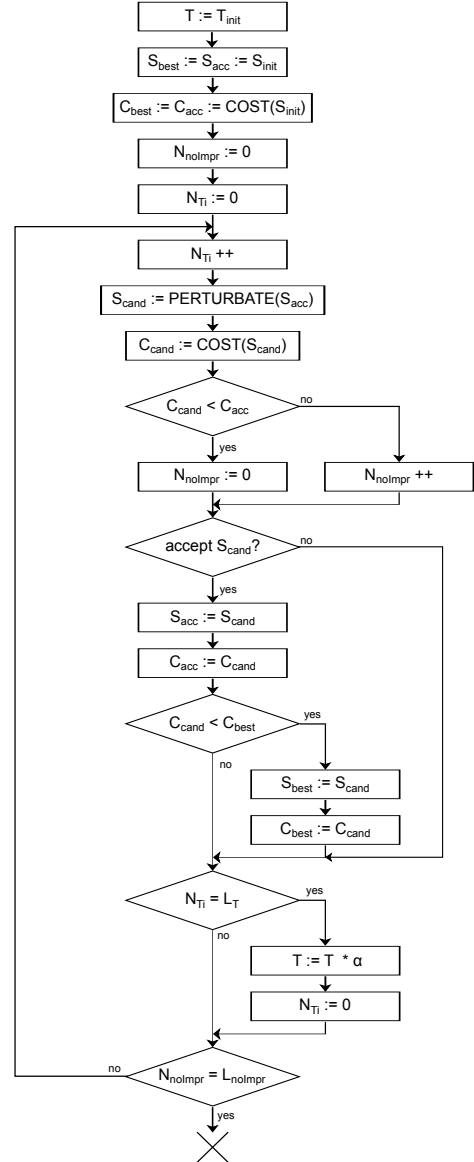


Figure 2 Problem-adapted simulated annealing algorithm

$C_{cand} \leq C_{acc}$ or a random number drawn from a uniform distribution between 0 and 1 is less than P_A according to equation (1), we set S_{cand} as the new accepted solution S_{acc} . If also $C_{cand} < C_{best}$, we additionally set it as the best encountered solution S_{best} .

If a fixed number of iterations at the current temperature (temperature length L_T) is reached, we reduce the temperature according to equation (3). At this point, we integrated an optional self-tuning mechanism not shown in figure 2: If the fraction of accepted candidates during the first L_T iterations is lower than a threshold, it increases the temperature. The normal annealing procedure then starts from this increased temperature.

We terminate the algorithm if $C_{noImpr} = L_{noImpr}$, i. e. the

last L_{noImpr} candidate solutions did not improve the cost C_{acc} of the accepted solution. These solutions were either rejected or probabilistically accepted despite higher cost. The latter likely results in a subsequent improvement. Hence, not finding any improvement is a good indication that the algorithm has converged to a (local) optimum that it is unlikely to leave.

4.3.1 Perturbation

The perturbation algorithm is supposed to derive a neighbor solution, i. e. a solution with minor modifications. With equal probability, our algorithm either randomly removes one bypass from the candidate solution, or it adds one unidirectional bypass between a random pair of nodes (which are not connected by a link of the basic topology or an existing bypass). For simplicity, we opted for this approach rather than a more fine-grained one which additionally extends or reduces the length of existing bypasses. Our approach also better reflects the vicinity of configurations if a make-before-break principle is applied.

4.3.2 Demand Routing and Cost Computation

We determine the cost of a candidate solution in five steps. First, the demands are routed deterministically into the network. Thereby, we determine the traffic load on each link and the amount of transit traffic (in the upper layer) across all nodes (R_{transit}) as continuous rate values. Geographically, the demands follow their shortest path in the basic topology (which depends on the edge weights). Along this path, they use the combination of bypasses which results in the least number of hops (i. e. nodes where electrical processing is necessary). If there are several combinations with this minimal hop count, an arbitrary one is chosen.

Second, we compute the required number of optical circuits on each link and in total (N_{circuits}) from the traffic load. In order to assure a minimum connectivity in the network, we assume at least one active circuit on each link of the basic topology (even if it carries no traffic).

Third, we determine the required number of transponders at each node from the number of circuits it terminates. This step is not required for the actual optimization but to finally evaluate the hardware requirements.

Fourth, we establish the number $N_{\text{switching}}$ of circuit set-ups and tear-downs by comparing the number of circuits between each node pair with the previous configuration.

Finally, we compute the total cost of the solution as the weighted sum of the total number of circuits, the amount of transit traffic and the number of switching operations:

$$C_{\text{solution}} = c_{\text{circuit}} \cdot N_{\text{circuits}} + c_{\text{processing}} \cdot R_{\text{transit}} + c_{\text{reconfig}} \cdot N_{\text{switching}} \quad (4)$$

where c_{circuit} is the energetic cost of one optical circuit, $c_{\text{processing}}$ is the energetic cost per traffic unit of switching

traffic electrically (in one node), and c_{reconfig} is the reconfiguration penalty per circuit set-up or tear-down. For the latter, the sensible value range is $0 \leq c_{\text{reconfig}} < c_{\text{circuit}}$, since larger values would favor maintaining unused circuits and thus contradict the idea of dynamic bypassing. We suggest setting the ratio of c_{circuit} and $c_{\text{processing}}$ according to the relation of the actual energy consumption on the two network layers, which depends on the respective technology. In the upper layer, for instance, IP routing would consume more energy than MPLS label switching.

5 Evaluation

For a first evaluation of our dynamic bypassing algorithm, we implemented an event-driven simulation tool based on the Java edition of the IKR Simulation Library [15]. In order to be able to control the statistical outcome of the optimization heuristic, we chose deterministic, periodically varying traffic demands.

In the following, we first present the simulation scenario. We then address the speed of convergence and the quality of the results depending on the parameters of the simulated annealing algorithm. We finally shed light on the performance of our bypassing scheme in terms of energy efficiency and network stability by evaluating metrics similar to [11].

5.1 Scenario

5.1.1 Network Topology

We base the evaluations on the Atlanta reference network available from SNDLib [16]. Figure 3 depicts its topology. It consists of 15 nodes and 22 bi-directional links, leaving room for 166 directed bypasses. We route the traffic along the geographically shortest paths, which also have the minimal number of hops.

5.1.2 Traffic Demands

We apply a complete and uniform demand matrix, i. e. the directed demand values between all 210 ordered pairs of distinct nodes are identical at a given time. These demand values follow a sinusoidal day profile, where the minimum demand is 25 % of the peak value. Since we assume that the network is reconfigured every 15 minutes, we extract 96 equidistant samples from the demand curve.

In the following, we specify demand and traffic values relative to the capacity of one circuit (in *circuit equivalents*). The performance of dynamic bypassing essentially depends on the ratio between the traffic demands and the circuit granularity. In order to investigate this effect, we scale the demand profiles such that their peak value varies between 0.1 and 2 circuit equivalents. We refer to the scaled demand curves by this peak value.

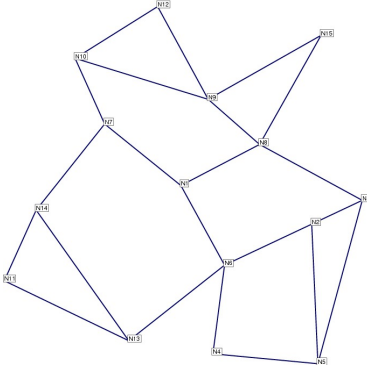


Figure 3 Atlanta network topology

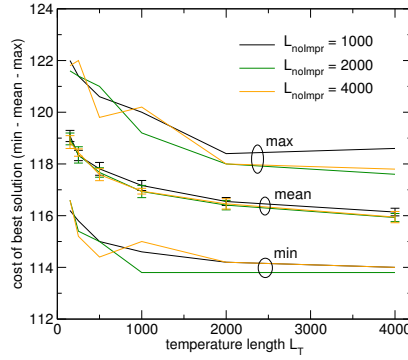


Figure 4 Cost of best solutions for demands of 0.2 circuit equivalents

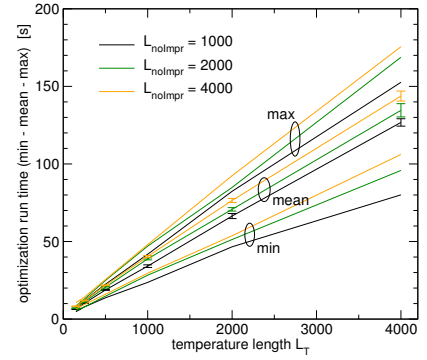


Figure 5 Optimization execution time for demands of 0.2 circuit equivalents

5.1.3 Algorithm Parameters

The parameters required by the algorithm are the cost factors and the cooling schedule. Since only the ratios of the three cost factors are of interest, we normalize $c_{\text{circuit}} = 1$. We chose the energetic cost of switching one circuit equivalent in the upper layer to be equal to the cost of one circuit: $c_{\text{processing}} = c_{\text{circuit}}$. Considering one traffic flow, this means that the break-even point for using a circuit bypassing one node is only reached if this bypass circuit is fully utilized. However, if further traffic also uses one basic topology link, setting up a one-node bypass is worthwhile as soon as the traffic on this link exceeds one circuit equivalent.

We vary the reconfiguration penalty within its sensible range in order to investigate its effect. We consider values of $c_{\text{reconfig}} = 0, 0.25, 0.5, 0.75$.

For simulated annealing, the appropriate initial temperature strongly depends on the cost difference of two neighbor solutions, and thus on the network topology, traffic demands, and cost parameters. For the considered scenario, a value of $T_{\text{init}} = 2.0$ proved to meet a target initial acceptance ratio of 0.8 in the majority of cases. We enforce this initial acceptance ratio by enabling the self-tuning mechanism. The cooling factor is set to $\alpha = 0.95$. We determine the temperature length and the iteration count for the termination condition experimentally in section 5.2. For the subsequent study, we choose $L_T = 1000$ and $L_{\text{noImpr}} = 2000$.

5.2 Optimization Performance

In order to investigate the impact of the cooling schedule parameters on the optimization performance and execution time, we solve the same problem instance with different parameter settings. In addition, we need to account for the statistical variations of the outcome due to the stochastic solution procedure. For this, we repeat the optimization 120 times with a differently seeded random number generator for each parameter set.

We discuss the results for the Atlanta network and a uniform demand value of 0.2 circuit equivalents. Figure 4

plots the cost of the found solution over the temperature length L_T for different termination condition values L_{noImpr} . The three groups of curves give the minimum, mean, and maximum of the 120 optimization runs per parameter set. All plots exhibit a trend of decreasing improvement with increasing temperature length. The different L_{noImpr} values do not produce a clear trend, except for a slight improvement of the mean and a partial improvement of the maximum when moving from 1000 to 2000. The extremal values lie within a corridor of $\pm 3\%$ of the mean. Since optimization runs with a much slower cooling schedule did not find a solution of a cost below 113.6 (which is very close to the minima of figure 4), we suspect that this is the global optimum. Under this assumption, the mean cost values of figure 4 deviate less than 5% from optimality, and the worst-case observation less than 8%. We consider these margins sufficient for an execution-time optimized algorithm.

Figure 5 gives the execution time of a single-threaded implementation of the algorithm on a state-of-the-art server CPU depending on the cooling schedule. Again, minimum, mean, and maximum of the 120 runs per parameter set are given. All plots show an approximately linear increase with the temperature length L_T , which is plausible if we assume that the termination condition effectively depends on the temperature. The increase of the execution time with L_{noImpr} is also plausible.

To select the temperature length, we have to trade off the quality of the solution against the execution time. We identify $L_T = 1000$ as a good compromise and choose $L_{\text{noImpr}} = 2000$. With these parameters, the algorithm converges in less than 50 seconds, which is more than sufficient given the assumed network reconfiguration interval.

5.3 Bypassing Effects

In the following, we discuss the effect of our bypassing algorithm on the energy consumption and the reconfiguration effort assuming the diurnal traffic pattern. We control for

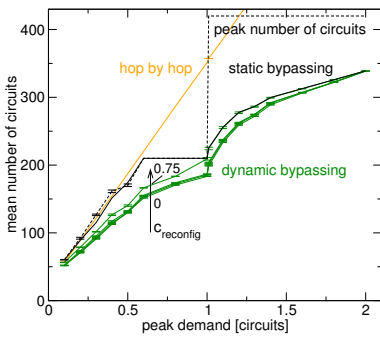


Figure 6 Mean number of active circuits in the network

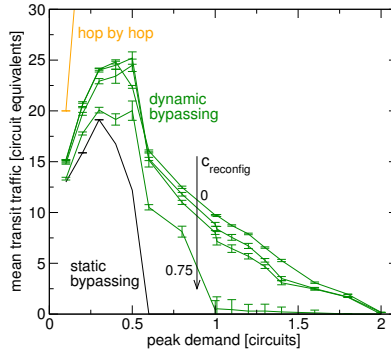


Figure 7 Mean amount of transit traffic in the network

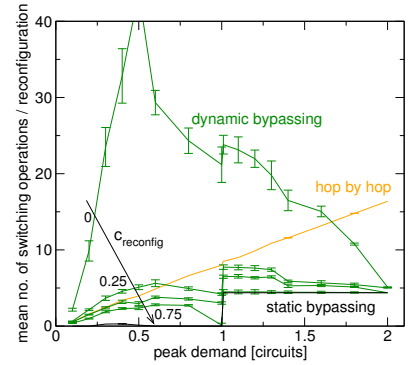


Figure 8 Mean number of switching operations per reconfiguration

the statistical variations of the optimization results by averaging over ten day periods.

As baseline, we also consider: (i) a static bypassing scheme with a bypass configuration optimized for the peak load; (ii) hop-by-hop routing in the basic topology without any bypasses. Both use fixed topologies, but the number of circuits per link is dynamically adapted to the traffic load.

5.3.1 Number of Active Optical Circuits

Figure 6 plots the average number of active circuits in the network over the peak demand. For very low load, all curves converge to the minimal 44 circuits of the basic topology. For hop-by-hop routing, the number of circuits increases approximately linearly with the load.

As expected, all bypassing schemes show a slower increase. Their curves exhibit a trend to saturate as the peak demand approaches 1 circuit equivalent, and increase more strongly for demands exceeding 1, before saturating again. The static bypassing curve, which essentially corresponds to the maximum number of circuits of the dynamic schemes for peak demands up to 1, suggests an explanation: As soon as the uniform demand value exceeds 0.5 circuit equivalents, the most efficient configuration is a full mesh of circuits (which requires 210 circuits in the Atlanta network). In this case, a circuit cannot carry two demands entirely, thus any link confronted with more than one demand would need at least two circuits. Due to the additional cost of processing in the upper layer, establishing a circuit for each demand is then more efficient.

The increase of the mean number of circuits for the dynamic bypassing schemes for peak demands above 0.5 essentially reflects the increasing share of time the actual demand is greater than 0.5 and the full mesh persists. If the demand exceeds 1, a second circuit is required on all links of the full mesh (since we do not split demands). Again, the time share of this condition governs the mean number of circuits. Since it also adapts the number of circuits per link in the full-mesh topology, the static bypassing scheme

performs similarly to the dynamic schemes. This effect is partly due to the uniformity of the demand matrix.

Reconfiguration penalties of $c_{\text{reconfig}} \leq 0.5$ have little effect on the mean number of circuits. $c_{\text{reconfig}} = 0.75$ results in a higher number of circuits, particularly in an interval around a peak demand of 1. While the penalty cannot prevent the establishment of circuits when the demands rise, it does hinder their tear-down when demands shrink. Consequently, the full-mesh configuration persists for longer periods of time. For peak demands greater than 1, the full mesh turns permanent (hence the convergence to the static bypassing scheme).

The dashed curve finally gives the peak number of circuits for dynamic bypassing with $c_{\text{reconfig}} = 0.5$. It gives a good approximation of required hardware resources in terms of transponders. Obviously, uniform demands exceeding 1 circuit double the number of transponders. An interesting effect is observed for peak demands below 0.5: Due to varying bypass configurations, more transponders may be required than for static bypassing. However, considering resource restrictions in our algorithm would likely eliminate this effect.

5.3.2 Amount of Transit Traffic

Figure 7 gives the time average of the amount of transit traffic switched in the upper layer in all nodes. Initially, this amount increases with the load, and it does so in a continued manner for hop-by-hop routing. For the bypassing schemes, the increasing establishment of bypass circuits inverts this trend. For static bypassing, the transit traffic drops to zero for peak demands above 0.5. This confirms the interpretation that a full mesh of circuits is established. For a peak demand of 2 (i.e. diurnal demand variations between 0.5 and 2), this condition applies to all bypassing schemes. Reciprocally to the number of circuits, the transit traffic decreases with increasing c_{reconfig} (due to different scales of the ordinates, the effect appears disproportionate in figures 6 and 7).

5.3.3 Number of Switching Operations

Figure 8 plots the mean number of switching operations in the network per network reconfiguration event (i. e. per re-optimization every 15 minutes) over the load. It clearly shows the benefit of a non-zero reconfiguration penalty for network stability. Further increasing a positive c_{reconfig} still has some effect, but at a much lower scale.

Static bypassing incurs switching operations if the peak load exceeds 1 circuit. These operations occur when the time-dependent demands rise above or fall below 1, and a second circuit is established or torn down on all links. Hence, the moderate mean value of 4.4 switching operations conceals the simultaneous modification of 210 circuits. However, this synchronization is an artifact of the demand uniformity. Due to the large number of circuits it involves, hop-by-hop routing incurs more switching operations than most bypassing schemes for high load.

6 Conclusion

In this paper, we proposed a dynamic optical bypassing algorithm based on simulated annealing. In view of periodic network reconfiguration in the order of 15 minutes, the algorithm is designed to minimize execution time while providing reasonably efficient bypass settings.

We evaluated this algorithm on a 15-node reference network. It produced good network configurations (presumably within 8 % of optimality) in convergence times below 50 seconds. In addition, we showed that setting a reconfiguration penalty effectively reduces the number of circuit set-ups and tear-downs without significant impact on the energy consumption.

In the considered scenario of uniform demands, dynamic bypassing did however not significantly outperform a static bypassing scheme with load-dependent adaptation of optical circuits. Future work should therefore investigate the benefit of using the dynamic bypassing algorithm for more realistic traffic demands. In addition, the scalability of the bypassing algorithm for larger network topologies needs to be analyzed. Moreover, the deterministic demand routing strategy of the algorithm could be extended. For instance, traffic could be rerouted to avoid lowly utilized additional circuits on bypass links. In addition, a restricted form of demand routing optimization could be included into the optimization procedure.

Acknowledgments

The author would like to thank Joachim Scharf for valuable discussions and Mirja Kühlewind for feedback and editorial support.

The work described in this paper was carried out with the support of STRONGEST, an Integrated Project funded by the European Commission through the 7th ICT-Framework Programme under grant agreement No. 247674.

References

- [1] J. Baliga, R. Ayre, *et al.* Energy Consumption in Optical IP Networks. *Journal of Lightwave Technology*, 27(13):2391–2403, July 2009.
- [2] DE-CIX traffic statistics. <http://www.de-cix.net/content/network/Traffic-Statistics.html>.
- [3] B. Mukherjee. *Optical WDM Networks*. Optical Networks. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [4] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [5] R. Durán, I. de Miguel, *et al.* Genetic algorithm to design logical topologies in reconfigurable WDM networks. *Photonic Network Communications*, 17:21–33, 2009.
- [6] J. Kuri, N. Puech, and M. Gagnaire. Diverse routing of scheduled lightpath demands in an optical transport network. In *Proceedings of 4th Intl. Workshop on Design of Reliable Communication Networks (DRCN)*, pages 69–76, oct. 2003.
- [7] S. Bjørnstad, D. R. Hjelle, and N. Stol. A highly efficient optical packet switching node design supporting guaranteed service. In *Proceedings of ECOC 2003*, pages 110–111, September 2003.
- [8] F. Idzikowski, S. Orłowski, *et al.* Dynamic routing at different layers in IP-over-WDM networks – Maximizing energy savings. *Optical Switching and Networking, Special Issue on Green Communications*, (3):181–200, July 2011.
- [9] Y. Zhang, M. Murata, *et al.* Traffic-based reconfiguration for logical topologies in large-scale WDM optical networks. *Journal of Lightwave Technology*, 23(10):2854–2867, 2005.
- [10] P. N. Tran and U. Killat. Distributed algorithm for dynamic logical topology reconfiguration in IP over WDM networks. In *Proceedings of ISCC 2009*, pages 748–756, 2009.
- [11] J. Scharf. Efficiency analysis of distributed dynamic optical bypassing heuristics. In *Proceedings of Workshop on Green Communications and Networking at ICC 2012*, June 2012.
- [12] T. F. Gonzalez, editor. *Handbook of Approximation Algorithms and Metaheuristics*. Chapman & Hall/CRC Computer & Information Science Series. Chapman & Hall/CRC, 2007.
- [13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [14] E. Aarts and J. K. Lenstra. *Local search in combinatorial optimization*. J. Wiley, New York, 1997.
- [15] IKR Simulation Library. <http://www.ikr.uni-stuttgart.de/Content/IKRSimLib>.
- [16] S. Orłowski, M. Pióro, *et al.* SNDlib 1.0–Survivable Network Design Library. *Networks*, 55(3):276–286, 2010. <http://sndlib.zib.de>.