FINDING TRANSPORT NETWORK CONFIGURATIONS USING SUPERVISED MACHINE LEARNING

Tobias Enderle

Institute of Communication Networks and Computer Engineering, University of Stuttgart, Stuttgart, Germany
tobias.enderle@ikr.uni-stuttgart.de

Keywords: INTEGER LINEAR PROGRAMMING, MACHINE LEARNING, SOFTWARE DEFINED NETWORKING, SUPERVISED LEARNING, WIDE AREA NETWORKS

Abstract

We propose a machine learning-based approach using a random forest for the fast computation of optimized transport network routing configurations. An evaluation in a 7-node network shows that our approach achieves competitive results in terms of solution quality and computation time compared to an exact ILP solution.

1 Introduction

The worldwide network traffic is constantly increasing [1] and new technologies like 5G will change the dynamic characteristics of this traffic fundamentally. New services emerging from use cases like enhanced mobile broadband or the connected car will be spatially and temporally much more volatile than traffic of today's network services. Network operators have to adapt their networks to those new characteristics and traffic amounts. One component in this network adaption process is the move to more efficient and flexible traffic routing schemes. Reconfiguring the traffic routes in the network on a regular basis allows the network operator to handle peak hours and low load situations as well as volatile traffic requirements efficiently [2]. The increasing "softwarization" of communication networks [3, 4] together with more flexible optical transmission technology [5, 6] allows for a reconfiguration of traffic routes in short time intervals. To gain benefits from short-term reconfigurations, the computation of new network configurations needs to be fast, too. An optimal configuration can be computed using optimization approaches like integer linear programming. However, since routing optimization problems are difficult to solve, the required solution times are hard to estimate and tend to be high. Another drawback of classical optimization is that it can only hardly benefit from the fact that traffic situations are repeating over time with only small variations. In these cases, similar traffic situations have to be optimized independently although often they lead to similar network configurations.

To overcome these limitations alternative approaches are necessary. Machine learning is a promising alternative to tackle those shortcomings. Machine learning-based approaches typically consist of a training and a prediction phase. Most of the computation time has to be invested in the training phase whereas the predictions are made very quickly. This allows the deployment in online reconfiguration engines. Furthermore, machine learning models are able to generalize to some extent, i.e., once trained with a certain training set they can handle new and unseen data that is similar to those in the training set as well without additional retraining.

Different machine learning applications in the field of communication networks have been developed recently. Many focus on physical layer problems like quality of transmission (QoT) estimation or non-linearity mitigation, but there are also various works on network layer topics like traffic prediction, failure management or flow classification [7]. Furthermore, the authors of [8] realize path and wavelength selection for an optical burst switched network using a machine learning approach while the authors of [9] have developed a machine learning algorithm for the routing and wavelength assignment (RWA) problem in optical networks.

In this paper we present a machine learning-based approach for the computation of transport network configurations. During network operation, the proposed method replaces the classical optimization solver with a machine learning classifier. Only during the training phase the solver is used to train the machine learning model with optimal network configurations for different traffic matrices. During network operation the model predicts optimal or near-optimal network configurations quickly allowing a network operator to trigger reconfigurations in very short time intervals. In contrast to previous works, we consider a routing problem that takes traffic grooming and network hardware of the electrical layer into account and minimizes the number of required router line cards in a realistic and dynamic traffic scenario.

2 Problem Statement

We consider a transport network consisting of a circuit-oriented optical layer and a packet-oriented electrical layer on top which could be realized by an IP-over-DWDM architecture (Fig. 1). We assume that the network is controlled by a software-defined networking (SDN) controller with global knowledge on all network state. The physical topology is given as graph $G = (V,E)$, where V is the set of nodes and E is the set of fibre

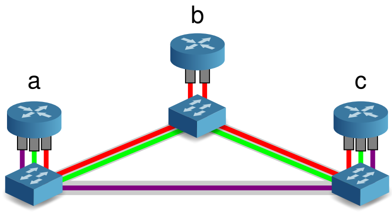

Fig. 1 Transport network architecture example consisting of packet routers and optical switches. The nodes are connected through different optical circuits (violet, green, red).

Fig. 2 Logical fibre topology of the 7-node network.

links. Each node consists of an optical switch and a packet router connected to it. The optical switch could be realized by a contentionless, directionless and colourless multi-degree reconfigurable optical add-drop multiplexer (ROADM) [10]. Therefore, arbitrary optical circuits can be established between any fibre, which connects to another node, and any of the ports in the router line cards.

Between any two nodes $s \in V$ and $t \in V$ ($s \neq t$) there exist two directed traffic demands $h_{s,t}$ and $h_{t,s}$ which describe the aggregated data rates that need to be provisioned between the two nodes. We denote the set of all demands as demand matrix H. Demands traverse the network on optical circuits and can be groomed if a circuit has enough capacity to hold multiple demands. An optical circuit requires one line card port at the router of its source node and one at the router of its target node. Intermediate nodes are bypassed, i.e., the circuit is switched in the ROADM without intermediate termination in the electrical domain. We require that the length of an optical circuit is within the transparent reach.

Using an integer linear program (ILP) similar to the one used for the resource minimization in an earlier publication of ours [11], we can find an optimal network configuration R, consisting of a set of optical circuits assigned to each demand in H, such that the amount of required router line cards is minimized. In the following we propose a machine learning-based approach that replaces the ILP solver during network operation. It is trained to predict the same optimal configuration R while using less computation time than the ILP.

3 Solution Approach

In our solution approach we treat the routing problem as a classification problem in which a machine learning algorithm assigns a network configuration to a given traffic situation. Hence, we consider a set of traffic demands H as input and the corresponding network configuration R as classification output. This means that one unique configuration defines one class and configurations with exactly the same circuits belong to the same class. We employ a supervised learning methodology, i.e., we train the classifier with a pre-computed training set $S = \{(H_1, R_1), (H_2, R_2), \ldots\}$ in which we known the optimal network configuration R_i for each set of traffic demands H_i. We test the quality of the classifier with a separate test set T. For the generation of the training set we employ an ILP that selects an optimal traffic route for each demand out of a pre-computed set of candidate paths [11]. A candidate path for a demand from node s to t is defined by a base path, i.e., a path in the physical topology G that connects s and t, and a sequence of one or multiple optical circuits along that base path. An example is given in Fig. 1 where different candidate paths for a demand from a to c are shown. Two base paths exist, first the direct path using fibre link a–c and secondly, the path via node b using the links a–b and b–c. On the direct base path only a direct optical circuit from a to c (violet) is possible, resulting in one candidate path. On the second base path, a direct circuit (green) and a sequence of two circuits (red) with intermediate termination in node b are possible, resulting in two candidate paths. In total 3 candidate paths exist in this example. As stated above, we don't consider circuits that exceed the transparent reach.

Due to the detailed routing information in each network configuration the number of classes as well as the output dimension in this classification problem tend to be very large. Different works [12, 13] have shown that *random forests* are a powerful machine learning approach which outperform other machine learning algorithms in many situations and in particular in situations with high-dimensional data [13]. For this reason, and because the number of parameters that need to be tuned for this type of algorithm is small, we use a random forest in our solution approach. As presented in [14], a random forest is an ensemble of decision trees. Each decision tree is a binary tree consisting of a simple numerical decision in each inner node which is evaluated based on the input data. By traversing the tree a leaf node is reached which assigns its candidate class to the input data. In a random forest each decision tree is trained on a random subsample of the training data. The prediction of the random forest is a combination of the decisions of the individual decision trees. A prediction is made very quickly because it requires the evaluation of a fixed number of binary decisions only.

4 Results

We evaluated our approach in a 7-node network which was derived from the Abilene topology found in the SNDlib [15]. The logical fibre topology is depicted in Fig. 2. We assumed an IP-over-DWDM setup utilizing router line cards that can hold up to 6 tunable ports [16]. Each port transmits at a data rate of 150 Gbps and has a transparent reach of 1800 km. The available

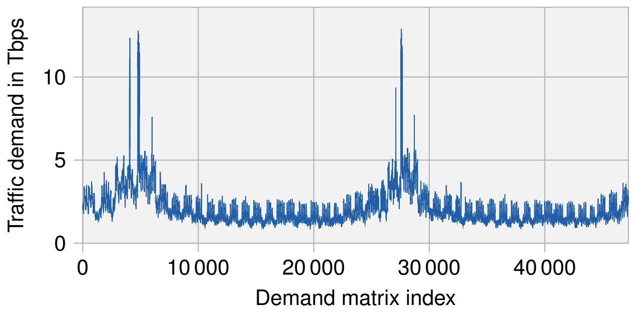

Fig. 3 Total traffic demand per demand matrix.

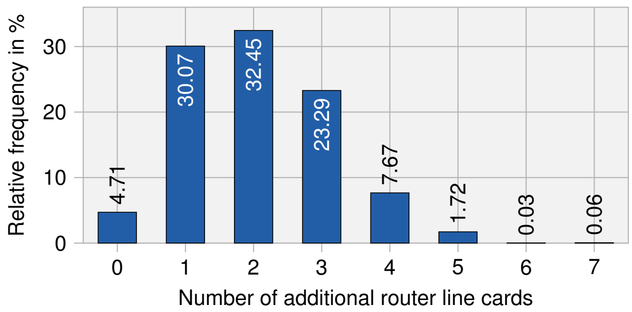

Fig. 4 Histogram for the number of required additional line cards for incorrect predictions compared to the ILP optimum.

traffic data for this topology consist of 48 096 matrices and is based on actual measurements [15]. In order to obtain a classifier that is not biased towards individual traffic matrices which appear several times in the original traffic data we removed all duplicates, which results in 47 232 unique traffic matrices. Fig. 3 shows the total amount of traffic in each matrix which has been scaled to an average demand of 50 Gbps per node pair. We computed optimal network configurations for each traffic matrix using the ILP that minimizes active router line cards and used 40 000 data points as training set S and the remaining 7232 data points as test set T. Since the topology consists of 7 nodes a single traffic matrix H contains 42 directed demands and, therefore, the input dimension of the random forest is 42. Since one route per demand is selected by the random forest its output dimension is 42 as well. For the whole dataset of 47 232 demand matrices the ILP found 10 037 unique optimal network configurations. The implication is that different traffic matrices map to the same network configuration which in turn allows the machine learning algorithm to generalize. Since a unique network configuration defines a class the number of classes the random forest has to distinguish equals 10 037.

For the described scenario the classifier achieves an accuracy, i.e., a proportion of correctly predicted configurations among the test cases in the test set, of 51.85 %. For the remaining 48.15 % of incorrectly classified test cases at least one path in the predicted configuration is different from the optimal paths in R_i. Yet, the random forest predicts 84.42 % of the individual paths in R_i correctly on average. Hence, even though a predicted configuration as a whole is not optimal, the majority of paths is still equal to the paths in the optimal configuration. Consequently, even the test cases that are not predicted correctly can provide decent network configurations. To investigate this further, we study the quality of the incorrect predictions in terms of the required router line cards. The optimal number of required line cards averaged over the test set is 7.83 while the maximum is 48. Fig. 4 depicts a histogram for the number of additional line cards required for the incorrectly predicted configurations compared to the ILP optimum. For 4.71 % of the incorrect predictions the random forest approach does not require any additional router line cards. In the worst case 7 additional line cards are necessary. Four additional line cards are sufficient for more than 98 % of the incorrect cases. Averaged over the whole test set, i.e. correct and incorrect predictions, 0.99 additional line cards are necessary.

Using CPLEX 12.8 on one core of a 3.4 GHz machine, the computation times required for the ILP range from time spans below one second up to more than a minute depending on the particular input traffic matrix. This is not much because the network consists of 7 nodes only. However, the computation times will rise significantly for larger topologies. In contrast, the prediction time of the random forest is independent of the particular input traffic matrix and only depends on the tree depth and the number of trees in the forest. Using the same machine as for the ILP and the scikit-learn 0.19.2 library [17] for the implementation of the random forest, the prediction times of the random forest were consistently below one second.

5 Conclusion

In this paper we have presented a machine learning-based approach for the computation of transport network configurations. We employ a supervised learning methodology based on a random forest to replace traditional solution approaches for the optimization of network configurations like ILPs. The evaluation in a 7-node network has shown that the new approach is able to provide network configurations with only 0.99 additional line cards on average compared to the ILP optimum. In 51.85 % of the test cases the output of the random forest is equal to the optimum. Four additional line cards are sufficient in 98 % of the cases in the whole test set. In contrast to an ILP solver, the computation times of the random forest approach are independent of the particular input demand matrix. This is increasingly important when the network size grows. The presented approach is a viable alternative to classical solution methods in the evaluated scenario. With increasing network sizes, however, the complexity of the classification in terms of the required amount of training data and time will grow as well. Future research will have to show whether the approach can handle these cases as well.

6 Acknowledgements

This work has been performed in the framework of the CELTIC EUREKA project SENDATE-TANDEM (Project ID C2015/3-2), and it is partly funded by the German BMBF (Project ID 16KIS0460K). The authors alone are responsible for the content of the paper.

7 References

[1] Cisco: 'Cisco Visual Networking Index: Forecast and Trends, 2017–2022', 2018

[2] Birk, M., Choudhury, G., Cortez, B., et al.: 'Evolving to an SDN-Enabled ISP Backbone: Key Technologies and Applications'. IEEE Communications Magazine, 2016, 54, (10), pp. 129–135

[3] Tse, S., Choudhury, G.: 'Real-Time Traffic Management in AT&T's SDN-Enabled Core IP/Optical Network'. 2018 Optical Fiber Communications Conference and Exposition (OFC), San Diego, CA, USA, March 2018, pp. 1–3

[4] Huang, T., Yu, F.R., Zhang, C., et al.: 'A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges'. IEEE Communications Surveys & Tutorials, 2017, 19, (2), pp. 891–917

[5] Nelson, L.E., Zhang, G., Padi, N., et al.: 'SDN-Controlled 400GbE end-to-end service using a CFP8 client over a deployed, commercial flexible ROADM system'. 2017 Optical Fiber Communications Conference and Exposition (OFC), Los Angeles, CA, USA, March 2017, pp. 1–3

[6] Dupas, A., Layec, P., Verchere, D., et al.: 'Ultra-fast Hitless 100Gbit/s Real-Time Bandwidth Variable Transmitter with SDN optical control'. 2018 Optical Fiber Communications Conference and Exposition (OFC), San Diego, CA, USA, March 2018, pp. 1–3

[7] Musumeci, F., Rottondi, C., Nag, A., et al.: 'An Overview on Application of Machine Learning Techniques in Optical Networks'. IEEE Communications Surveys & Tutorials, 2018

[8] Kiran, Y.V., Venkatesh, T., Ram Murthy, C.S.: 'A Reinforcement Learning Framework for Path Selection and Wavelength Selection in Optical Burst Switched Networks'. IEEE Journal on Selected Areas in Communications, 2007, 25, (9), pp. 18–26

[9] Martín, I., Hernández, J.A., Troia, S., et al.: 'Is Machine Learning Suitable for Solving RWA Problems in Optical Networks?'. 2018 European Conference on Optical Communication (ECOC), Rome, Italy, September 2018, pp. 1–3

[10] ITU-T G.672: 'Characteristics of multi-degree reconfigurable optical add/drop multiplexers', 2018

[11] Bauknecht U., Enderle T., Witt A.: 'Reduction of Delay Overfulfillment in IP-over-DWDM Transport Networks'. 23rd Conference on Optical Network Design and Modelling (ONDM), Athens, Greece, May 2019

[12] Abraham, B., Mandya, A., Bapat, R., et al.: 'A Comparison of Machine Learning Approaches to Detect Botnet Traffic'. 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, July 2018, pp. 1–8

[13] Caruana, R., Karampatziakis, N., Yessenalina, A.: 'An Empirical Evaluation of Supervised Learning in High Dimensions'. Proc. of the 25th International Conference on Machine Learning, Helsinki, Finland, July 2008

[14] Breiman, L.: 'Random forests'. Machine learning, 2001, 45, (1), pp. 5–32

[15] Orlowski, S., Wessäly, R., Pióro, M., et al.: 'SNDlib 1.0–Survivable Network Design Library'. Proc. of the 3rd International Network Optimization Conference (INOC), Spa, Belgium, April 2007

[16] 'Cisco Network Convergence System 5500 Series: 1.2-Tbps IPoDWDM Modular Line Card Data Sheet', https://www.cisco.com/c/en/us/products/collateral/routers/network-convergence-system-5500-series/datasheet-c78-739372.html, accessed 29 March 2019

[17] Pedregosa, F., Varoquaux, G., Gramfort, A., et al.: 'Scikit-learn: Machine learning in Python'. Journal of Machine Learning Research, 2011, 12, (Oct), pp. 2825–2830