A Generic Algorithm for CAPEX-Aware Multi-Layer Network Design

Michael Duelli, Eduard Weber, and Michael Menth
University of Wuerzburg, Institute of Computer Science, Chair of Distributed Systems
Am Hubland, 97074 Würzburg, Germany
E-mail: {duelli|weber|menth}@informatik.uni-wuerzburg.de

Abstract

Capital expenditure (CAPEX), i.e. equipment cost, is a decisive criterion for the design of a new network infrastructure. The design of efficient transport networks requires the interconnection of multiple technologies which form separate layers with their own networking view and routing. This leads to the notion of *multi-layer networks*. CAPEX-aware design of such networks requires algorithms which propose multi-layer hardware configurations that are able to carry a given set of traffic demands at minimal CAPEX. We present a generic algorithm for CAPEX-aware multi-layer network design as well as a computationally viable implementation and perform evaluations on realistic network topologies. The underlying CAPEX and multi-layer technology models are explained in detail.

1 Introduction

Ongoing exponential growth in Internet traffic, decreasing revenues, and the emergence of new services, like Carrier Ethernet, trigger network providers to upgrade and optimize their network configurations.

The basis of most carrier networks is optical fiber which connects *points of presence* (PoP), called *sites*, over large distances. Several technologies must be installed at a site to provide different services like data transmission on the IP/MPLS, Ethernet, or SDH/SONET protocol. Each offered service and its technology form a separate *layer* which has its own networking view. These separate views lead to the notion of *multi-layer networks*.

To achieve cost-efficiency, *multi-layer network design* has to respect all available technologies, their features, and economics. This requires detailed *capital expenditure* (CAPEX) models for multi-layer networking equipment which have been recently published [1, 2].

The level of detail of these CAPEX models allows for a more realistic network design, but also increases the runtime complexity of CAPEX-aware multi-layer network design. For parameters sets of this size, *integer linear programs* (ILP) tend to have long runtimes even for medium-sized network instances. Hence, we develop heuristics for CAPEX-aware multi-layer network design which can yield sub-optimal results within feasible time. Such heuristics are interesting for science as well as economy due to their ability to perform a plenty of parametric studies within a short time even for large network instances and detailed CAPEX models.

The authors have been funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (BMBF Förderkennzeichen 01BP0775). Their work is part of the EUREKA project “100 Gbit/s Carrier-Grade Ethernet Transport Technologies (CELTIC CP4-001)”. The authors alone are responsible for the content of the paper.

In this paper, we present a generic CAPEX-aware algorithm for multi-layer network design with a detailed CAPEX model and a computationally viable implementation which proves reasonable runtimes even for large network instances. We develop a software tool [3] to evaluate multi-layer network design algorithms.

The remainder of this paper is structured as follows. Section 2 introduces the problem formulation and gives an overview of related work. Section 3 explains the CAPEX and multi-layer technology models. In Section 4, the generic algorithm for multi-layer network design is presented along with an efficient implementation which is evaluated on several network instances in Section 5. Section 6 summarizes this paper.

2 Problem Formulation and Related Work

In this section, we formulate the considered optimization problem and give an overview of related work.

2.1 Problem Formulation

There mainly are two reasons to install multiple technologies at a site. On the one hand, the provision of multiple networking services, like the connectivity via *Internet Protocol* (IP) or Ethernet, requires the installation of different networking equipment per service. For instance, IP connectivity requires routers whilst switches are used for Ethernet. Hence, two technologies have to be installed to provide both connectivity services at a site.

On the other hand, the installation of multiple technologies can lead to lower CAPEX of a network. For example, the easiest way to set up a connection between remote sites is the usage of interfaces which can directly connect to the physical topology of a network and

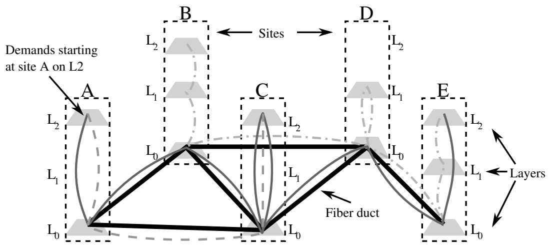


Figure 1: An abstract view on an exemplary multi-layer network with three demands.

transmit the signal over long-haul distances. The configuration of such *point-to-point* connectivity is simple, but the required long-haul interfaces are expensive and use the physical transmission medium exclusively. Instead of giving exclusive access, one or more intermediate *transport networks* can be used which allow to aggregate several data signals of arbitrary technologies. These intermediate technologies form separate *logical layers* with their own routing and serve as wrappers for the physical or underlying logical topology, respectively. Although this can increase cost-efficiency and resource utilization, the transport technology must be efficiently used to justify its initial acquisition.

An exemplary result of a multi-layer network design for a simple topology is illustrated in Figure 1. It shows the physical layer L_0 and two logical layers L_1 and L_2 as well as the routing of three demands which start at L_2 and partly use L_1 for transport.

In this paper, CAPEX-aware design of *multi-layer networks* means to interconnect multiple services (i.e. technologies) across remote sites by the installation of networking equipment to set up cost-efficient logical topologies upon the physical topology. Connections in a logical layer are realized as concatenations of connections in a lower layer. We use a detailed CAPEX model [1] to design a network that can carry a given set of traffic demands of a certain service upon an immutable physical topology which consists of optical technology. This optimization problem is approached by heuristics.

2.2 Related Work

The meaning of the term *multi-layer* slightly varies in publications on multi-layer network optimization. On the one hand, it is used for multiple abstract views on a single technology to separate available functionalities into logical layers. On the other hand, it is used for the existence of multiple interconnected technologies each forming a separate layer with its own routing.

The first group of papers focuses on the aggregation of data, so called *grooming*, in optical technology. In [4], a graph model is developed for traffic grooming that facilitates the change of the optimization objective and two algorithms for the optimization of these graphs are presented. The number of wavelength links and transponders is reduced to indirectly minimize network cost. The authors of [5] consider multi-layer switches in *optical transport networks* (OTN) which can deal with wavelengths, wavebands, and fiber as aggregation levels. Evaluations are performed for a simple non-linear cost function that maps the utilization of the used aggregation level. The problem is approached by an ILP as well as a heuristic which is faster, but less successful. In [6], an additional "grooming layer" is introduced and ILPs are given for the optimization of CAPEX that rely on pre-calculated paths to speed-up the optimization. A simple CAPEX model is used for the optimization to state the benefit of optical multiplexing equipment. In [7], we present ILPs that do not depend on pre-calculated paths and develop heuristics for this problem.

The second group of papers relies on (detailed) technology and CAPEX models to interconnect different technologies. The authors of [8] develop an ILP for a detailed equipment model but give no evaluations. In [9], they enhance their ILPs by heuristics within a branch-and-cut algorithm and focus on two-layer network design. In [10], a comparison of different networking architectures is presented by evaluating a dimensioning process for multi-layer networks, but no algorithms are given. In [11], we consider a multi-layer network design containing IP and OTN technology. The focus is on CAPEX optimization under shared risk link groups caused by single link failures in the physical network. In [12], we model a CAPEX optimization for transparent, semi-transparent, and opaque optical networks by ILPs in a canonical way.

Both groups of papers apply a *multi-layer network design* which means the creation of a network equipment configuration such that given traffic demands can be carried by a network. This makes it different from mere *multi-layer routing* which denotes the path finding process in a multi-layer network with an immutable network configuration. For instance, the authors of [13] consider the routing of traffic flows in multi-domain networks with multiple services that have to be compatibly interconnected. Their routing bases on the ITU-T recommendation G.805 elaborated in [14].

For all kinds of network design, there are two basic approaches concerning the premises of existing legacy equipment. First, a network can be designed from scratch merely respecting a couple of fixed parameters like site locations and physical adjacencies as in, e.g. [6, 7]. Second, an existing network can be migrated by updating and extending existing network technology as in, e.g. [8, 15]. The latter is more realistic but still lacks sufficient models which have to respect, e.g., moving hardware between sites, intentional violation of *service level agreements* (SLA), or financial budget restrictions for migration. In [15], several strategies for multi-period planning are analysed using the CAPEX model of [16] – the predecessor of [2] – and only includes optical equipment.

In this paper, we consider the CAPEX-aware design of multi-layer networks from scratch. We use the CAPEX model from [1] to interconnect several technologies while minimizing CAPEX. Due to the complexity of the CAPEX and technology model, fast heuristics are the only means to perform evaluations on network instances of realistic size within feasible time. Furthermore, fast heuristics allow to perform a plenty of parametric studies within a given time span. To the best of our knowledge, no heuristics have been published that are fast and cover multi-layer network design on a detailed CAPEX model.

3 Modeling

In this section, we describe a network technology model and the CAPEX model used for evaluation.

3.1 Network Technology Model

Networking equipment is constructed in a modular fashion. Hence, we strictly divide the equipment of all considered technologies into four component groups as in the CAPEX model of [2]. The component groups are: basic nodes, slot cards, port cards, and interfaces.

A *basic node* is the main component in each technology. It provides power supply as well as cooling for all components of this technology and a number of slots. In this model, a basic node deals with the switching of data and has a switching capacity.

A *slot card* of the same technology can be plugged into each slot of a basic node and also has a switching capacity. The sum of the switching capacities of the plugged-in slot cards must not exceed the maximum switching capacity of a basic node. Hence, a more powerful basic node must be installed if more overall switching capacity or slots are required. Every slot card provides a number of ports.

A *port card* of the same technology can be plugged into each port of a slot card and also has a switching capacity. The sum of the switching capacities of the plugged-in port cards must not exceed the maximum switching capacity of a slot card. If more switching capacity or ports are required, a further slot card has to be installed. In [2], some technologies have either slot cards or port cards. We generalize our model and provide all technologies with zero-cost dummy slot or port cards, respectively. Finally, port cards provide a number of connectors and there can be several types of port cards with different connectors.

An *interface* can be plugged into each connector of an adequate port card. Interfaces are the components that send or receive data using a certain encoding. If more connectors are required, a further port card has to be installed.

The principle of assembling these four component groups is illustrated in Figure 2.

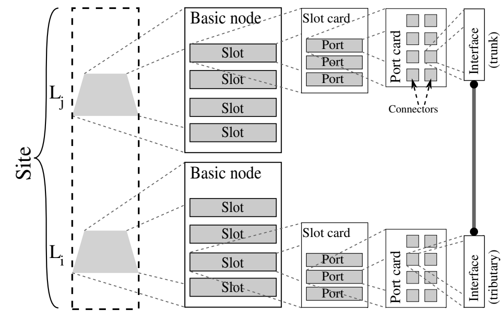


Figure 2: Setup of a connection within a site between two technology layers L_i and L_j.

3.2 Multi-Layer Interconnection Model

The assembly of the four component groups within a single technology is only restricted by switching capacities and numbers of slots, ports, or connectors. The setup of connections via interfaces across layers and sites underlies several technological constraints.

A connection can only be established between two *compatible* interfaces. Depending on their technology and ability, interfaces use a specific framing to en/decapsulate data. The framed data is transmitted on an interface specific bit rate, the interface's capacity. Two interfaces must use the same encoding of data, i.e. framing and capacity, to be compatible. Furthermore, the data encoding of an interface depends on two properties, its reach and aggregation capability. The *reach* of an interface is the transmission distance up to which an encoded signal can still be decoded. It can range from several to thousands of meters. For instance, cheap short-reach interfaces are sufficient for the interconnection of different technologies within a single site as depicted in Figure 2. The *aggregation capability* of an interface is used to multiplex several lower bit rate data flows and send it as a single higher bit rate data flow. In turn, a compatible interface must be used that can demultiplex this data at the sink. The principle of aggregation can be any multiplexing technique and depends on the technology.

Two kinds of interfaces are used to set up connections: *trunk interfaces* communicate data downwards in the layer hierarchy, whereas *tributary interfaces* communicate upwards. A compatible pair of trunk and tributary interfaces is needed to set up a connection between two layers L_i and L_j. This is illustrated in Figure 2 which shows a trunk interface in layer L_j that connects to a tributary interface in layer L_i while both are embedded in a valid hardware configuration of their layer. The hardware configuration for a connection across several layers and remote sites is depicted in Figure 3. Herein, a connection is set up from layer L_2 at site A to L_2 at site C using layer L_1 as a transport network which passes site B. As a consequence, there is a direct logical connection from site A to C on layer L_2 which is realized as the concatenation of the connections from

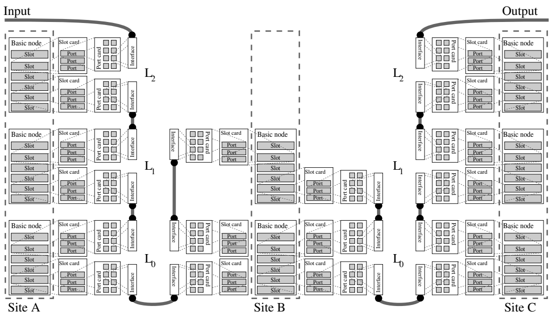


Figure 3: Setup of a connection across remote sites via multiple technology layers.

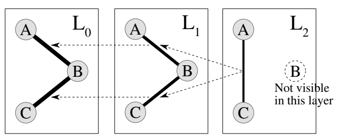


Figure 4: An abstract view on the topologies resulting from the hardware interconnection in Figure 3.

site A to B and from site B to C on layer L_1 while each of these two connections is realized as direct connection on the physical layer L_0. Figure 4 illustrates the topologies in each layer resulting from this hardware configuration.

In our model, all data is finally transmitted on layer L_0 which consists of bundles of fibers, so called *fiber ducts*. The ducts physically connect remote sites.

3.3 CAPEX Model

So far, we have described the modular setup and interconnecting of network technology. The CAPEX model not only associates cost values with equipment, but also defines which network equipment is available and possible interconnections. We use the CAPEX model of [1] which respects the modular structure of network equipment described in Section 3.1. It contains IP/MPLS, Ethernet, SDH/SONET, and OTN technologies which currently are widely deployed. Furthermore, the cost values in this model are vendor-independent and normalized to the cost of a 10 Gbit/s WDM transponder. A list of all available network equipment and possible interconnections in this model is given in [2], so we focus on its main properties in this paper.

In this model, all interfaces have a capacity of either 2.5, 10, or 40 Gbit/s which corresponds to the data rate ODU-1, ODU-2, and ODU-3, respectively, used in OTN networks. There are two kinds of aggregation techniques: *wavelength division multiplex* (WDM) and *time division multiplex* (TDM). The WDM technique can be used in OTN networks to transmit 40 or 80 wavelengths via a single fiber in parallel. We only use 40 wavelengths for the evaluations in this paper. The TDM technique transmits data flows divided in fixed time slots as a single higher bit rate data flow and is available in SDH/SONET by *virtual containers* (VC), and in OTN by muxponders. In the CAPEX model, two kinds of TDM are defined: multiplexing four 2.5 Gbit/s data flows to one 10 Gbit/s and multiplexing four 10 Gbit/s data flows to one 40 Gbit/s data flow. Statistical multiplex of packet-switched services, like IP and Ethernet, is not considered in this model.

In general, a higher capability leads to a higher CAPEX in this model. For instance, a higher reach, a higher capacity, or the ability to aggregate data increases the CAPEX of an interface and a higher switching capacity increases the CAPEX of a basic node. The model also reflects certain specifics of network equipment markets, e.g. the relative high cost of IP equipment. This information will be considered in future work.

We use the CAPEX model of [1] as a basis and slightly extend its equipment by zero-cost dummy components to fit in the four-components structure described in Section 3.1. Furthermore, we strictly separate the OTN technology into an *optical channel* (OCh) layer handling wavelengths and an *optical multiplex section* (OMS) layer dealing with bundles of wavelengths multiplexed on fibers.

Upon these multi-layer technology and CAPEX models, we develop algorithms for a CAPEX-aware design of multi-layer networks.

4 Algorithm

In this section, we introduce a generic algorithm for CAPEX-aware multi-layer network design and present an efficient realization, called *auxiliary cross layer* (AXL) algorithm.

4.1 Terminology

Our generic algorithm for CAPEX-aware multi-layer network design requires three input parameters: a physical network topology, a CAPEX model, and a set of traffic demands.

The immutable physical layer L_0 incorporates a network topology $\mathcal{G}(\mathcal{V}, \mathcal{E}_0)$ which contains a set of sites $\mathcal{V}$ and connects these sites by a set of fiber ducts $\mathcal{E}_0 \subseteq \mathcal{V} \times \mathcal{V}$. We denote the topology of a layer L_i by $\mathcal{G}(L_i) = \mathcal{G}(\mathcal{V}, \mathcal{E}_i)$ since the sites $\mathcal{V}$ are the same in all layers $L_i \in \mathcal{L}$, and each layer L_i contains all information on its edges $\mathcal{E}_i \subseteq \mathcal{V} \times \mathcal{V}$.

The CAPEX model $\mathcal{C}$ defines a set of available technologies $\mathcal{L}$ and their components. The possible interconnections of a layer $L \in \mathcal{L}$ given by the CAPEX model C are represented by a set of layers $\mathcal{L}(L) \subseteq \mathcal{L}$ containing all layers to which layer L can connect. As each layer L is connected to itself, $L \in \mathcal{L}(L)$ always holds.

We consider a multi-layer traffic matrix $\mathcal{D}$ of directed demands $d \in \mathcal{D} \subseteq \mathcal{V} \times \mathcal{V} \times \mathcal{L} \times \mathcal{B}$ each being a quadruple $(S(d), T(d), L(d), B(d))$ where $S(d), T(d) \in \mathcal{V}$ are the demand's source and target, respectively, while

Algorithm 1 GENERICALGORITHM

Input: Physical topology $\mathcal{G}(L_0)$,
CAPEX model $\mathcal{C}$, traffic demands $\mathcal{D}$
$\mathcal{R} = \emptyset$ {initialization}
for all $d \in$ SORTDEMANDS$(\mathcal{D}, \mathcal{G}(L_0), \mathcal{C})$ **do**
$\mathcal{R} = \mathcal{R} \cup$ FINDROUTINGS$(d, L(d), \mathcal{C})$
end for
Output: SELECTROUTINGS$(\mathcal{D}, \mathcal{R}, \mathcal{C})$

Algorithm 2 FINDROUTINGS

Input: Demand $d \in \mathcal{D}$, current layer $L \in \mathcal{L}$,
CAPEX model $\mathcal{C}$
$\mathcal{R} = \emptyset$ {initialization}
for all $L' \in \mathcal{L}(L)$ **do**
$\mathcal{R} = \mathcal{R} \cup$ ROUTE$(S(d), T(d), \mathcal{G}(L'))$
if $L' \neq L_0$ **then** {abort recursion}
$\mathcal{R} = \mathcal{R} \cup \{\mathcal{R}(d) :$ FINDROUTINGS$(d, L', \mathcal{C})\}$
end if
end for
Output: $\mathcal{R}$ {routings found in recursion}

$L(d) \in \mathcal{L}$ is the layer it starts from and $B(d)$ is its requested bandwidth.

The multi-layer network design algorithm creates edges in logical layers to route a traffic demand $d \in \mathcal{D}$. Such an edge $e \in \mathcal{G}(L_i)$ in a logical layer $L_i \in \mathcal{L}$ is recursively defined as a concatenation of edges in layers $L_j \in \mathcal{L}, 0 \leq j < i$, as depicted in Figure 4. Analogously, the routing $\mathcal{R}(d) \subseteq \mathcal{G}(L(d))$ of a demand $d \in \mathcal{D}$ is defined as a concatenation of edges in layer $L(d)$ between sites $S(d)$ and $T(d)$.

4.2 Generic Algorithm

Our multi-layer network design algorithm creates logical topologies $\mathcal{G}(L_i)$ upon the immutable physical topology $\mathcal{G}(L_0)$ such that all considered traffic demands $d \in \mathcal{D}$ can be carried by the network.

In Algorithm 1, we present a formal description of our generic multi-layer network design algorithm. The algorithm uses generic subroutines which can be exchanged. The algorithm sequentially processes the demands in their given order. Since, the order of the demands has an impact on the resulting multi-layer network structure and CAPEX, the demand order can be changed in SORTDEMANDS before the actual algorithm starts. Then, the subroutine FINDROUTINGS defined in Algorithm 2 is called for each demand. Basically, FINDROUTINGS tries to find a routing for a demand d within the currently considered layer L and also recurses in all layers $\mathcal{L}(L)$ that may be connected to L. The recursion ends when the considered layer is the physical layer which must contain a routing path for all demands. Otherwise, no connection between the requested source and target is possible.

The algorithm FINDROUTINGS yields a set of possible routings for a demand d by calling the subroutine ROUTE. The resulting routings for a demand d are added to the set $\mathcal{R}$ of all routings found by the algorithm. The routing algorithm ROUTE can be any single-layer routing algorithm that is applicable on the considered topology.

When all demands have been processed, the algorithm calls the subroutine SELECTROUTINGS in Algorithm 1 to select one routing per demand d in the set of routings $\mathcal{R}$. Since we strive for CAPEX-aware design of multi-layer networks, SELECTROUTING should consider the CAPEX of such routing selections.

The definition of the algorithm is intentionally kept generic to be able to control the trade-off between its runtime and accuracy.

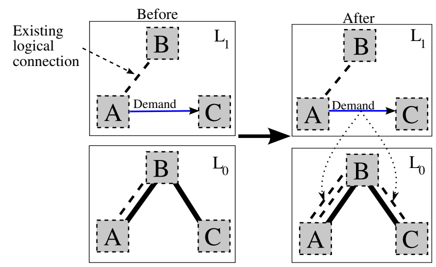


Figure 5: Algorithmic limitation due to a fixed view on layer L_1 and sequential demand routing.

Our algorithm can also use and re-configure legacy equipment. Nevertheless, we only consider empty logical layers in this paper.

4.3 Auxiliary Cross Layer Algorithm

For an efficient realization of Algorithm 1, it is neither feasible to evaluate all demand orders in the subroutine SORTDEMANDS nor to calculate all possible paths in subroutine ROUTE which would yield the optimum for sure. Instead, we introduce an efficient realization, called *auxiliary cross layer* (AXL) algorithm.

To keep the runtime of the AXL algorithm feasible, we introduce two fundamental restrictions. First, we do not change the demand order in $\mathcal{D}$, i.e. SORTDEMANDS is the identity function, and leaves the demand order unchanged. Second, we use the Dijkstra routing algorithm [17] for the subroutine ROUTE to find a routing in a topology. In the development of the AXL algorithm, we focus on relieving the consequences of using a non-optimal demand order.

The generic algorithm Algorithm 1 only examines the topology of a single layer L at a time in subroutine ROUTE. During the network design process, new edges are created in layer L sequentially. As a consequence, no path may exist that could be used to route a traffic demand $d \in \mathcal{D}$ in layer L at the time d is processed. In this case, the algorithm installs a new direct logical connection between $S(d)$ and $T(d)$ in layer L, but cannot make use of any existing connections in layer L. This is illustrated by Figure 5 in which a new direct connection is installed in layer L_1 between sites A

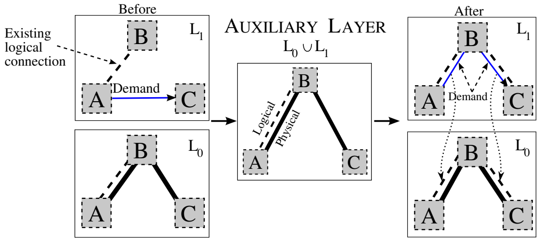
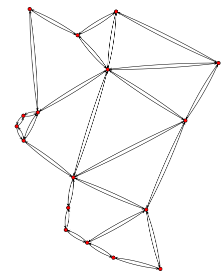
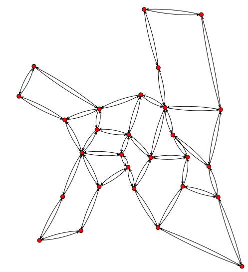


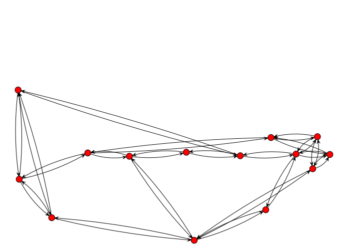
Figure 6: Construction and usage of an auxiliary layer to find cross layer paths.



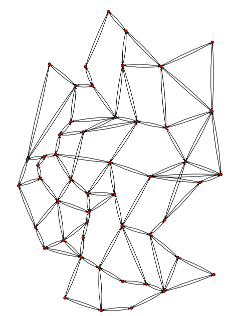
(a) Nobel Germany (G17) ($|\mathcal{V}| = 17, |\mathcal{E}_0| = 52$)



(b) Nobel Europe (EU) ($|\mathcal{V}| = 28, |\mathcal{E}_0| = 82$)



(c) Nobel U.S. (US) ($|\mathcal{V}| = 14, |\mathcal{E}_0| = 42$)



(d) Germany50 (G50) ($|\mathcal{V}| = 50, |\mathcal{E}_0| = 176$)

Figure 7: The four considered physical topologies from [18] with directed edges.

and C including resource usage in L_0 depicted by dotted lines. The existing connection in layer L_1 from site A to B and its resources in L_0 (dashed line) cannot be used since the algorithm is not able to recognize that only a connection between site B and C is required to set up a path between site A and C.

The limited view on a single layer L_j can be resolved by introducing the concept of an auxiliary layer L_{aux} that unites the edges of all logical layers below layer L_j, i.e. $\mathcal{G}(L_{\text{aux}}) = \cup_{0 \leq i \leq j} \mathcal{G}(L_i)$. This auxiliary layer is not part of the multi-layer network itself, but is used to find a routing considering several layers when calling the subroutine ROUTE. Figure 6 illustrates the construction of the auxiliary layer for the example shown in Figure 5. In the auxiliary layer for layers L_0 and L_1, a path can be found that uses the existing logical connection in layer L_1 and a multi-layer routing can be found that only needs to install one additional connection from site B to C in both layers. In general, the algorithm is able to find routings using several layers which is impossible merely considering a single layer.

Another shortcoming of the current algorithm is its inability to work around blocked network resources. Due to the sequential routing of the demands, a resource can be completely occupied by a single demand which blocks the routing of any further demands via this resource. For instance, a demand will be routed via a fiber without using WDM equipment, since it is not required for the routing of this demand and it is cheaper not to use the technology. As a consequence, no further demands can be routed using this fiber, the resource is blocked.

To decrease the number of blocked resources, we introduce a *grooming* mechanism, i.e. a routing optimization to achieve more efficient traffic transport. When no routing can be found for a demand, we are looking for resources that can be unblocked by using additional aggregation equipment on existing connections. Therefore, the algorithm considers paths that could be used for routing if there still was spare capacity and tries to aggregate the traffic on such candidate paths. The candidate paths are not only considered in a single layer, but also cross layer paths found via an auxiliary layer L_{aux} can be used. If a grooming process is successful, the routing of all traffic demands using the modified resources is changed and the additional demand is routed along this path. If no routing can be found for a demand at all, the demand is blocked.

5 Evaluation

In this section, we evaluate the implementation of the *auxiliary cross layer* (AXL) algorithm on four network instances from [18] which span intra- to inter-country networks from medium to large size and are depicted in Figure 7. We introduced directed edges in all four topologies as our models work on directed demands and network equipment. Hence, there is a pair of opposite edges between each pair of connected sites. For

Table 1: Performance of the algorithm

Topology	Traffic demands k-unif., (total/blocked)	Runtime (s)	CAPEX
G17	1-unif., (272/0)	4.55	4716.06
G17	3-unif., (816/0)	19.46	14935.20
G17	6-unif., (1632/0)	52.97	29243.58
G17	7-unif., (1904/23)	67.46	33411.49
G50	1-unif., 2450	172.73	51566.25
G50	2-unif., (4900/742)	615.40	87824.44
EU	1-unif., (756/0)	22.06	13792.95
EU	2-unif., (1512/0)	57.10	29412.93
EU	3-unif., (2268/250)	117.59	38928.05
US	1-unif., (182/0)	2.30	3290.74
US	3-unif., (546/0)	9.76	10101.28
US	10-unif., (1820/0)	55.41	30101.13
US	12-unif., (2184/128)	82.69	33835.67

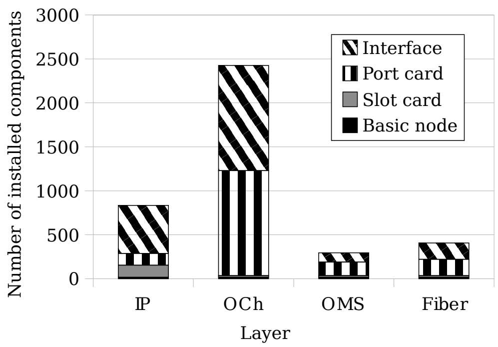


Figure 8: Study of installed components.

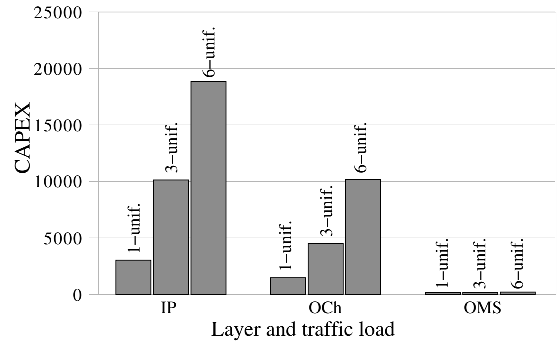


Figure 9: Study of traffic load.

the evaluation, we developed a graphical software tool [3] for multi-layer network design and used it for all evaluations in this paper.

We apply the AXL algorithm to these four network instances using the CAPEX and technology model from [2] which was described in Section 3 and includes the IP/MPLS, Ethernet, SDH/SONET, and OTN technology. We consider directed k-*uniform* traffic matrices with a fixed demand bit rate $\mathcal{B}$ of 10 Gbit/s (ODU-1). In this paper, traffic demands always start at the IP layer. A bit rate of 10 Gbit/s is realistic for today's core networks and, hence, other bit rates have not been considered in this paper. Furthermore, no networking equipment failures are considered and fiber ducts only contain a single fiber strain to show the performance of the algorithm with limited resources. All evaluations were performed on a Linux machine with Java 6 on an Intel(R) Core(TM)2 Duo CPU at 2.4 GHz using [3].

5.1 Runtime and Traffic Load

The results of a study on the algorithm's runtime and the amount of traffic it can route are compiled in Table 1. We used different values of k in k-uniform traffic matrices, i.e. there are k directed demands from each site to each other site on the IP layer. The runtime of the AXL implementation is below 11 minutes for all considered scenarios. The amount of traffic the algorithm can route depends on the distribution of the nodal degree in the physical topology graph which is the final limitation for routing any traffic. A full study of this dependency will be covered in future work.

While the resulting CAPEX of a network increases linearly with the amount of traffic, the runtime of the algorithm increases non-linear. This results from the computation time that has to be spent for the grooming mechanism described in Section 4.3 which is performed for every demand that cannot be routed without multiplexing previously routed demands. The computation time for the grooming mechanism increases with increasing traffic load since more demands have to be multiplexed. We further notice that several demands are blocked for higher traffic loads since resource utilization reaches its maximum.

5.2 Equipment and CAPEX Distribution

Besides the overall network CAPEX, our tool [3] yields a detailed view on the installed network equipment which allows insights on the structure of a multi-layer network. We use the *Nobel Germany* topology as an example for such evaluations with the AXL algorithm.

First, we analyse the number of installed components per technology split up into the four component groups defined by the CAPEX model in Section 3. Figure 8 shows the number of installed component that results from a 1-uniform IP traffic matrix. Interfaces are the most numerous group of components in the IP layers, since the considered traffic starts in this layer.

Second, we evaluate the impact of increasing network traffic on network CAPEX. Figure 9 shows the CAPEX for the *Nobel Germany* topology with 1-, 3-, and 6-uniform IP traffic split up by technology. It shows that the OMS layer is already fully equipped with 1-uniform traffic, but still has spare capacity for further wavelengths. In constrast, the CAPEX almost linearly increases for all other technologies up to 6-uniform IP traffic which was the maximum amount of traffic for the *Nobel Germany* topology as of Table 1. The Fiber layer which is shown in Figure 8 is not contained in Figure 9 since the components installed in this layer are only zero-cost dummy components introduced in our model and do not have an impact on CAPEX.

6 Summary

We introduced a generic algorithm for CAPEX-aware multi-layer network design. We formulated the multi-layer network optimization problem and presented models for CAPEX and multi-layer technology as well as the interconnection of such technologies.

We presented an efficient realization of the generic algorithm which is called *auxiliary cross layer* (AXL) algorithm and is implemented in a graphical software tool which is used to evaluate multi-layer network scenarios with multi-layer traffic. The AXL algorithm proves to find multi-layer network configurations even for large network topologies and high traffic load within minutes.

We showed the performance of the AXL algorithm by performing several evaluations with increasing network traffic on several network topologies and use these results to provide an initial evaluation of the CAPEX structure within the resulting network configurations.
In future work, we will extend the model of the generic algorithm and enhance its realization and implementation. Special focus will be on an improved demand order and reduced blocking rate of demands. Furthermore, we will take multi-layer traffic and multi-layer resilience into account.

Acknowledgements

The authors would like to thank Prof. Tran-Gia for the stimulating environment which was a prerequisite for this work.

References

[1] NOBEL Working Group, “NOBEL and NOBEL-2 Project Website,” Sep. 2008. [Online]. Available: www.ist-nobel.org

[2] R. Hülsermann, M. Gunkel, C. Meusburger, and D. A. Schupke, “Cost Modeling and Evaluation of Capital Expenditures in Optical Multilayer Networks,” *Journal of Optical Networking*, vol. 7, no. 9, pp. 814–833, Aug. 2008.

[3] *100GET-E3-R3G-2L (tool)*, Mar. 2009, www3.informatik.uni-wuerzburg.de/100GET.

[4] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, “A Novel Generic Graph Model for Traffic Grooming in Heterogeneous WDM Mesh Networks,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, 2003.

[5] G. Huiban, S. Pérennes, and M. Syska, “Traffic Grooming in WDM Networks with Multi-Layer Switches,” in *IEEE International Conference on Communications (ICC)*, vol. 5, 2002.

[6] M. Scheffel, R. G. Prinz, C. G. Gruber, A. Autenrieth, and D. A. Schupke, “Optimal Routing and Grooming for Multilayer Networks with Transponders and Muxponders,” in *IEEE Globecom* , San Francisco, CA, USA, Nov. 2006.

[7] M. Duelli, C. Pluntke, and M. Menth, “Minimizing Installation Costs of Survivable DWDM-Mesh Networks: A Heuristic Approach,” in 4^{th} *Conference on Next Generation Internet Design and Engineering (NGI)*, Krakow, Poland, Apr. 2008, pp. 15–22.

[8] S. Orlowski and R. Wessäly, “An Integer Programming Model for Multi-Layer Network Design,” Konrad-Zuse-Zentrum für Informationstechnik Berlin, ZIB Preprint ZR-04-49, Dec. 2004.

[9] S. Orlowski, A. M. Koster, C. Raack, and R. Wessäly, “Two-layer Network Design by Branch-and-Cut featuring MIP-based Heuristics,” Konrad-Zuse-Zentrum für Informationstechnik Berlin, Tech. Rep., 2006.

[10] M. Köhn, S. Bodamer, C. M. Gauger, S. Gunreben, G. Hu, and D. Saß, “Comparison of IP/WDM transport network architectures for dynamic data traffic,” in *11^{th} European Conference on Networks and Optical Communications (NOC2006)*, Berlin, Germany, 2006.

[11] M. Duelli, M. Hartmann, M. Menth, R. Hülsermann, and M. Düser, “Performance Evaluation of IP over Cost-Optimized Optical Multilayer Networks with SRLGs,” in *9. ITG-Fachtagung Photonische Netze*, Leipzig, Germany, Apr. 2008, pp. 21–29.

[12] C. Pluntke, M. Menth, and M. Duelli, “CAPEX-Aware Design of Survivable DWDM Mesh Networks,” in *IEEE International Conference on Communications (ICC)*, Dresden, Germany, Jun. 2009.

[13] F. Dijkstra, J. van der Ham, P. Grosso, and C. de Laat, “A Path Finding Implementation for Multi-Layer Networks,” *Future Generation Computer Systems*, vol. 25, no. 2, pp. 142–146, Feb. 2009.

[14] F. Dijkstra, B. Andree, K. Koymans, J. van der Ham, and C. de Laat, “A Multi-Layer Network Model Based on ITU-T G.805,” *Computer Networks*, Jun. 2007.

[15] C. Meusburger, D. A. Schupke, and J. Eberspächer, “Multiperiod Planning for Optical Networks - Approaches Based on Cost Optimization and Limited Budget,” in *IEEE International Conference on Communications (ICC)*, Beijing, China, May 2008.

[16] M. Gunkel, R. Leppla, M. Wade, A. Lord, D. A. Schupke, G. Lehmann, C. Fürst, S. Bodamer, B. Bollenz, H. Haunstein, H. Nakajima, and J. Martensson, “A Cost Model for the WDM Layer,” in *Photonics in Switching*, Creete, Greece, Oct. 2006.

[17] E. W. Dijkstra, “A Note on Two Problems in Connexion with Graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[18] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, “SNDlib 1.0–Survivable Network Design Library,” in *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, Apr. 2007, http://sndlib.zib.de.