Maximization of Single Hop Traffic with Greedy Heuristics

Esa Hyytiä
Networking Laboratory, HUT, Finland,
email: esa@netlab.hut.fi,

ABSTRACT
Maximization of the single hop traffic has been proposed as an computationally feasible approximation to the logical topology design problem in wavelength routed networks. In this paper we consider a class of greedy heuristic algorithms for solving the maximization of the single hop traffic problem. The considered heuristics configure one lightpath at a time onto the network. Two new variations of the previously proposed heuristic algorithm are presented and shown to be superior to the earlier heuristic algorithm by numerical simulations.
KEY WORDS
WDM, RWA, logical topology design, single hop traffic

1 Introduction

Wavelength division multiplexing (WDM) is a way to exploit efficiently the vast capacity of optical fibres. In all-optical wavelength routed (WR) networks the routing in the optical layer is based on wavelength channels offering transparent optical pipes through the network [1, 2]. Such networks are very suitable for the high capacity backbone networks where they are currently being deployed.

The optical layer provides a logical topology (LT) for a higher layer protocol, e.g. ATM or IP, where each lightpath constitutes a logical link. Eventually the trend is towards IP-over-Optical solutions, where IP packets are transferred directly on the optical layer without any intermediate layer. For example in IETF work is going on for standardizing the so called generalized multi protocol label switching (GMPLS), which is supposed to unify the management of the optical networks and allow interoperability between different manufacturers. A WR optical network is an attempt to use the best of both the optical and electronical world. The optical layer provides enormous capacity, while the electronical layer allows much higher granularity. In the logical topology design (LTD) the goal is to find an optimal configuration for both layers.

The topic has been studied a lot and in the literature one can find several formulations for the LTD problem, see e.g. [3–7]. Typically one optimizes some quantity like congestion in the network, the average packet delay or the total number of electronical interfaces. Usually the problem is formulated as a mixed integer linear programming (MILP) problem. The proposed formulations tend to lead to intractable problems, thus justifying the use of heuristic algorithms at the same time. The usual heuristic approach is to solve the problem in parts, first decide some set of lightpaths which constitute links for the logical layer, and then trying to find a feasible configuration for both the optical and the logical layer satisfying the LT constraints and minimizing the chosen objective function (e.g. minimizing the maximum load in logical links).

In [8] the authors proposed to maximize the single hop traffic as an alternative optimization goal. By single hop traffic we mean traffic flows that reach their final destination in one logical hop, i.e. there is a direct lightpath from the source node to the destination node. This simplifies the joint problem considerably by allowing one to neglect the traffic routing in the logical layer. In the same paper the authors proposed a simple greedy heuristic algorithm, called *CP1*, to solve the single hop maximization problem. The algorithm configures one lightpath at a time between such nodes where the volume of traffic on single hop increases the most. In this paper we propose two alternative greedy heuristics. The first heuristic algorithm, iterative *CP1*, is a combination of *CP1* and any static RWA algorithm. The static RWA algorithm is used to "pack" the current lightpaths more efficiently when needed in order to allow *CP1* to configure additional lightpaths. The second heuristic algorithm, denoted with *CP1e*, is similar to *CP1* but uses a novel dynamic order in which it configures the lightpaths. Both proposed algorithms are shown to be superior to *CP1*.

The rest of the paper is organized as follows. First, in section 2, we present one MILP formulation to the LTD problem and describe the respective terminology. Then, in section 3 the maximization of single hop traffic is formulated and explained together with a greedy heuristic algorithm proposed in [8] and two new greedy heuristics. In section 4 the new heuristics are shown to have a better performance. Finally, section 5 contains the conclusions.

2 MILP Formulation

In this section we give an MILP formulation for the joint LTD problem where the objective is to minimize the average number of optical hops a packet traverses. The used notation and variables, following [3, 4, 9], are presented in table 1. With these definitions the optimization problem can be formulated as follows:
Objective: minimize the average number of hops, i.e.,

$$\min \frac{1}{\sum_{(s,d)} \lambda^{(sd)}} \sum_{(i,j)} \lambda_{ij}, \tag{1}$$

Subject to:

Logical Topology Design:

$$\text{(links in)} \quad \sum_j b_{ji} \le \Delta^{(\text{in})}_{\max},\ \forall\, i \tag{2a}$$

$$\text{(links out)} \quad \sum_j b_{ij} \le \Delta^{(\text{out})}_{\max},\ \forall\, i \tag{2b}$$

$$\text{(value range)} \quad b_{ij} \in \{0, 1, 2, \ldots\},\ \forall\, (i,j) \tag{2c}$$

Traffic Routing: (logical layer)

$$\text{(congestion)} \quad \lambda_{ij} \le \lambda_{\max} \cdot b_{ij},\ \forall\, (i,j) \tag{2d}$$

$$\text{(total flow)} \quad \lambda_{ij} = \sum_{(s,d)} \lambda^{(sd)}_{ij},\ \forall\, (i,j) \tag{2e}$$

$$\text{(existance)} \quad \lambda^{(sd)}_{ij} \le \lambda^{(sd)} b_{ij},\ \forall\, (i,j),\ (s,d) \tag{2f}$$

(flow conservation)

$$\sum_j \lambda^{(sd)}_{ij} - \lambda^{(sd)}_{ji} = \begin{cases} \lambda^{(sd)}, \text{if } i = s, \\ -\lambda^{(sd)}, \text{if } i = d, \\ 0, \text{otherwise.} \end{cases} \ \forall\, (s,d),\ i \tag{2g}$$

$$\text{(value range)} \quad \lambda^{(sd)}_{ij} \ge 0,\ \forall\, (i,j),\ (s,d) \tag{2h}$$

Routing and Wavelength Assignment: (optical layer)

$$\text{(ch. assignment)} \quad \sum_k c^{(k)}_{ij} = b_{ij},\ \forall\, (i,j) \tag{2i}$$

$$\text{(consistency)} \quad c^{(k)}_{ij}(l,m) \le c^{(k)}_{ij},\ \forall\, (i,j),\ (l,m),\ k \tag{2j}$$

$$\text{(distinct channel)} \quad \sum_{ij} c^{(k)}_{ij}(l,m) \le p_{lm},\ \forall\, (l,m),\ k \tag{2k}$$

(lightpath cont.)

$$\sum_{k,m} c^{(k)}_{ij}(l,m) - c^{(k)}_{ij}(m,l) = \begin{cases} b_{ij}, \text{if } l = i, \\ -b_{ij}, \text{if } l = j, \\ 0, \text{otherwise.} \end{cases} \ \forall\, (i,j),\ l \tag{2l}$$

$$\text{(hop constraint)} \quad \sum_{lm} c^{(k)}_{ij}(l,m) \le h_{ij},\ \forall\, (i,j),\ k \tag{2m}$$

$$\text{(value range)} \quad c^{(k)}_{ij} \in \{0,1\},\ \forall\, (i,j),\ k \tag{2n}$$

$$c^{(k)}_{ij}(l,m) \in \{0,1\},\ \forall\, (i,j), (l,m), k \tag{2o}$$

2.1 Decomposition of LTD

The MILP formulation presented is often intractable and one must try other approaches to tackle with the problem. A straightforward approach is to solve the joint LTD problem in parts. Namely, we can decompose the problem into the following subproblems (see fig. 1):

i) Logical topology design (LTD), i.e. Eqs. (2a)-(2c).
ii) Traffic routing (TR), i.e. Eqs. (2d)-(2h).
iii) Lightpath RWA, i.e. Eqs. (2i)-(2o).

Basically, one can first fix some LT and then try to find a feasible RWA and TR for it. This can be repeated iteratively; the LT can be modified based on the current configuration and then steps ii) and iii) can be repeated.

constant	**explanation**
p_{ij}	**number of physical fibres** $i \longrightarrow j$, 0 if none.
$\lambda^{(sd)}$	**average traffic load** from s to d, traffic matrix in logical layer, e.g. pkt/s.
h_{ij}	**physical hops constraint**, the number of links a lightpath $i \longrightarrow j$ can traverse (constraint).
$\Delta^{(\text{in})}_{\max}$	**maximum logical in degree**, i.e. number of optical receivers.
$\Delta^{(\text{out})}_{\max}$	**maximum logical out degree**, i.e. number of optical transmitters.
$\lambda_{\max}$	**maximum congestion** in the logical layer, $\lambda_{\max} \le \max_{i,j} \lambda_{ij}$.
variable	**explanation**
b_{ij}	**number of lightpaths** $i \longrightarrow j$.
$c^{(k)}_{ij}$	**number of lightpaths** $i \longrightarrow j$ **using the wavelength channel** k.
$c^{(k)}_{ij}(l,m)$	**number of lightpaths** $i \longrightarrow j$ **using the wavelength channel** k **at link** $l \longrightarrow m$.
$\lambda^{(sd)}_{ij}$	**proportion of traffic** from s to d routed through lightpath $i \longrightarrow j$.
λ_{ij}	**virtual traffic load** in lightpath $i \longrightarrow j$, consists of proportions $\lambda^{(sd)}_{ij}$.

Table 1: Notation.

3 Maximization of Single Hop Traffic

One possibility to simplify the MILP formulation is to consider only the total volume of traffic carried with one optical hop as proposed in [8]. That is, we assume that one is only interested in the traffic which reaches its destination without electronical processing in intermediate nodes. Then we do not have to determine the routes in the logical layer, which simplifies the problem considerably. In this case the constraints (2d)-(2h) defining the traffic routing (TR) can be replaced with,

$$\text{(single hop traffic)} \ \lambda_{\text{one}} = \sum_{(s,d)} \lambda^{(sd)} \cdot b_{sd} \tag{3}$$

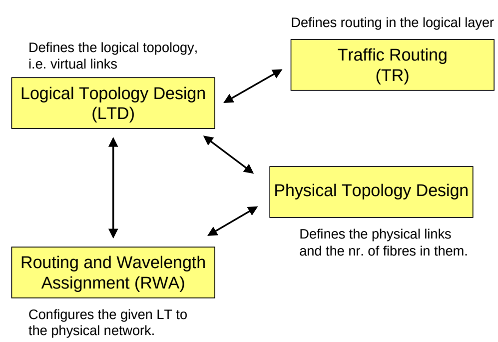

Figure 1: Relationships between the different subproblems. The global optimum requires taking each subproblem into account.

And as a new objective we try to maximize the single hop traffic, i.e. the objective function is simply,

$$\max \lambda_{\text{one}},$$

which is clearly a linear function. Note that this formulation still leaves the traffic routing (TR) in the logical layer open but fixes the logical topology (LT) and routing and wavelength assignment (RWA). Thus, solving the "single hop maximization" problem gives a logical topology for which one must solve, exactly or approximately, the traffic routing problem.

This approach of decomposing the joint problem was first proposed in [8], where the MILP formulation assumed a fixed set of one shortest path for each node pair. The authors named the problem as *CP* problem. We will return to this formulation and present respective heuristic algorithms later in section 3.1. When configuring lightpaths onto a network in order to maximize the volume of single hop traffic one can use the following principle:

Principle 1 (Configure fully used direct lightpaths first) *Let t_{sd} be the traffic flow $s \to d$ and let t'_{sd} denote the traffic which requires exclusively the total bandwidth of one lightpath, i.e.,*[1]

$$t'_{sd} = \lceil t_{sd} \rceil .$$

It is clearly advantageous to route the traffic corresponding to the t'_{sd} directly to its destination if possible. This can be seen as a traditional static RWA problem, where the goal is to minimize the number of used wavelengths. These lightpaths are already fully used and thus can be neglected when routing the remaining traffic in the logical layer.

Thus, the solution of RWA problem can be used as a starting point for any heuristics. The remaining problem is to configure the traffic matrix,

$$t^{(r)}_{sd} = t_{sd} - t'_{sd},$$

where each component is less than one. Typically the small traffic flows are combined at some node (traffic grooming) and routed further together.

3.1 *CP1* Heuristics

In [8] the authors formulated the single hop traffic maximization problem as an MILP problem with some additional restrictions (e.g. the routes were fixed beforehand), which was named as *CP* problem. Even though the *CP* problem is considerably easier to solve than the whole joint problem, the formulation can still lead to an intractable problem when the number of the network nodes is large. Therefore, the authors also suggested an approach where lightpaths are established on one wavelength layer at a time and named the respective problem as *CP1*. After configuring one layer the traffic matrix is updated by subtracting the traffic carried on the current layer. Then the same step can be taken for the next layer, if available. The authors also suggested a heuristic algorithm to solve the *CP1* problem, i.e. a greedy heuristic algorithm which assigns lightpaths at the current wavelength layer in the order defined by the (residual) traffic matrix.

Note that this order agrees with principle 1, i.e. the algorithm configures first such lightpaths $s \to d$, which will be used solely by the single-hop traffic $s \to d$. Similarly, once no more lightpaths can be established to the current layer the traffic matrix is updated and the algorithm moves to the next layer. This is repeated until the maximum number of wavelength layers is reached. Formally the heuristic algorithm is described in algorithm 1.

Algorithm 1 Heuristic Algorithm for *CP1* [8]

let $\mathcal{N}$ be the set of network nodes
set $t_{s,d} \leftarrow \lambda^{(sd)}$ {the initial traffic intensity $s \to d$}
$\mathcal{X} \leftarrow$ (one of) the shortest path ℓ for each node pair (s, d)
set $W \leftarrow 0$
while $W < W_{\max}$ **do**
    set $W \leftarrow W + 1$
    sort $\mathcal{X}$ in the descending order of traffic intensity $t_{s,d}$
    for each $\ell \in \mathcal{X}$ **do**
        if path ℓ is free at layer W **then**
            assign a lightpath to path ℓ at layer W
            $t_{s,d} \leftarrow \max\{0, t_{s,d} - C\}$ {remaining traffic}
        end if
    end for
end while

3.2 Iterative *CP1*

The *CP1* heuristics can be improved by using a static RWA algorithm to "pack" the current LT when the *CP1* algorithm is no longer able to configure more lightpaths. A good (and fast) candidate for static RWA problem, the layered RWA algorithm, is presented in appendix A. If the used static RWA algorithm manages to find more "space", then the *CP1* heuristics can continue and configure additional lightpaths. This can be repeated until no new lightpaths can be configured onto network.

The idea is simple and is formally presented in algorithm 2. As the iterative version only configures additional lightpaths to the solution(s) of the basic *CP1*, the amount of single hop traffic will never be less than what *CP1* alone

Algorithm 2 Iterative *CP1*

configure lightpaths using *CP1*
repeat
    reconfigure LT using static RWA algorithm ("pack")
    if more than W wavelength layers is used **then**
        return the previous legal configuration
    end if
    continue with *CP1*
until no new lightpaths can be configured

[1] or $\lceil t_{sd} + \delta \rceil$ where δ is some small constant.

would manage to configure. Note that as the RWA algorithm only configures lightpaths between the given set of $s-d$ pairs, the solution will never violate the logical degree constraints as long as the original solution is feasible.

3.3 Enhanced *CP1*

In algorithm 3 an enhanced version of *CP1* algorithm is presented. The main difference is that *CP1e* uses n shortest paths instead of one. Also the order in which paths are configured is slightly different and resembles closely the ideas behind the layered RWA algorithm 4. The order is defined by the first different number in the sequence (smaller first),

$$(\,\Delta(p),\ -\min\{1,t_{i,j}\},\ -\ell_p,\ p_1,\ \ldots,p_{n(p)}),$$

where $\Delta(p)$ is the number of additional hops the path p uses when compared to minimum possible, $t_{i,j}$ is the residual traffic from i to j, ℓ_p is the length of path p (in hops), and p_i are the node numbers along the path. Note that the term corresponding to the residual traffic matrix is modified to be $\min\{1,t_{i,j}\}$ to reflect the fact that one wavelength can carry at most one unit of traffic (i.e. it does not matter which connection is configured as long as the lightpath channel is fully utilized).

The fact that algorithm 3 configures the longer paths first instead of shorter paths may lead to worse overall results when the problem itself is ill-posed in that the available resources are inadequate with regard to the traffic demand. Otherwise, configuring the longer lightpaths first seems to be a good strategy as they are clearly harder to configure than the shorter lightpaths at later steps.

Note that for each layer the order is dynamic unlike the case is with the *CP1* algorithm. Furthermore, as the order of the paths reminds closely the order used in the layered RWA algorithm 4 the iterative version using algorithm 4 to reconfigure the current LT is unlikely to give any improvement. However, using a more sophisticated RWA algorithm may turn out to be successful.

Algorithm 3 Enhanced *CP1*

$\mathcal{X} \leftarrow n$ shortest routes for each $s-d$ pair
let $t_{i,j}$ be the (residual) traffic from i to j
let $z(p) = (\,\Delta(p),\ -\min\{1,t_{i,j}\},\ -\ell_p,\ p_1,\ \ldots,p_{n(p)})$
$W \leftarrow 1$
while $W \leq W_{\max}$ **do**
  $\mathcal{X}' \leftarrow \mathcal{X}$
  repeat
    take path $p \in \mathcal{X}'$ with the smallest $z(p)$
    if path p is free at layer W **then**
      configure a lightpath p at layer W
      update: $t_{i,j} \leftarrow \max\{0, t_{i,j}-1\}$
    end if
  until $\mathcal{X}'$ is empty
  $W \leftarrow W+1$
  $\mathcal{X} = \{p \in \mathcal{X} \ : \ t_{p_1,p_{n(p)}} > 0\}$
end while

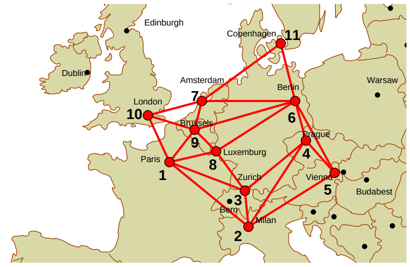

Figure 2: Cost 239 project test core network.

3.4 Connectivity

The problem with blindly maximizing the single hop traffic is that solution does not necessarily result in a connected logical topology. In the literature several approaches have been suggested to guarantee a connected logical topology. In [8] it is proposed that before filling the last wavelength layer one makes sure that the graph is connected by adding necessary lightpaths. On the other hand, in [3] and [9] it is proposed that if the logical degree is greater than the physical degree then one initially assigns one lightpath to each physical link.

4 Numerical LTD Example

In this section we present some numerical results obtained with the different heuristic LTD algorithms. As a test network we use the example network of the Cost 239 project, which is depicted in fig. 2. The traffic matrix, scaled to the capacity of wavelength channel, is

$$\mathbf{T} = \frac{1}{4}\cdot\begin{pmatrix} 0 & 5 & 6 & 1 & 2 & 11 & 5 & 1 & 7 & 10 & 1 \\ 5 & 0 & 6 & 1 & 3 & 9 & 2 & 1 & 2 & 3 & 1 \\ 6 & 6 & 0 & 1 & 3 & 11 & 3 & 1 & 3 & 3 & 1 \\ 1 & 1 & 1 & 0 & 1 & 2 & 1 & 1 & 1 & 1 & 1 \\ 2 & 3 & 3 & 1 & 0 & 9 & 1 & 1 & 1 & 2 & 1 \\ 11 & 9 & 11 & 2 & 9 & 0 & 8 & 2 & 6 & 8 & 3 \\ 5 & 2 & 3 & 1 & 1 & 8 & 0 & 1 & 4 & 5 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 & 1 & 0 & 1 & 1 & 1 \\ 6 & 2 & 6 & 1 & 1 & 6 & 4 & 1 & 0 & 4 & 1 \\ 10 & 3 & 3 & 1 & 2 & 8 & 5 & 1 & 4 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 3 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Heuristic algorithms *CP1*, *CP1e* and the iterative version of *CP1* were run for $W = 4, 6, 8, 10, 12$ and 14 wavelength channels. The number of available transmitters and receivers was assumed to be infinite. Note that routing at the logical layer was not determined, but instead the traffic carried in single hop was used as a performance criterion.

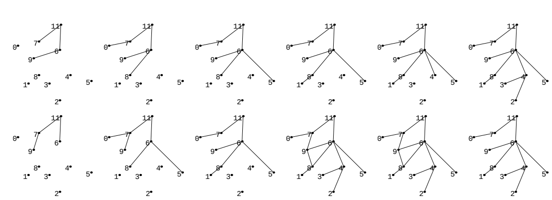

Figure 3: Configured lightpaths from Copenhagen (node 11). Upper row corresponds to the algorithm *CP1* with $W = 4, 6, 8, 10, 12$ and 14 wavelength channels. Lower row corresponds to the iterative version.

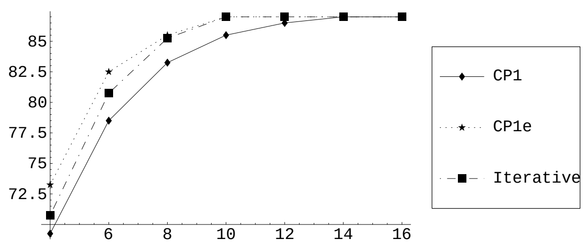

Figure 4: The volume of single hop traffic. On the x-axis is the number of wavelength channels and y-axis represents the volume of single hop traffic.

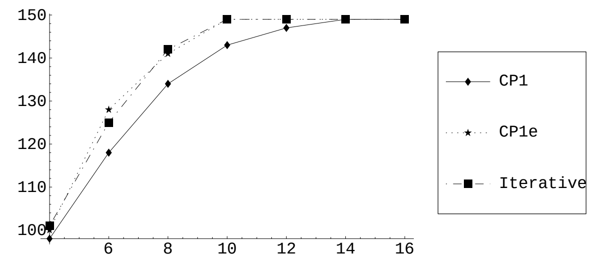

Figure 5: The number of configured lightpaths (logical links). On the x-axis is the number of wavelength channels and on y-axis the number of lightpaths.

In fig. 3 the configured lightpaths starting from Copenhagen are depicted for *CP1* and its iterative version as a function of available wavelength channels. On the first row are the lightpaths *CP1* has configured, while the second row contains the lightpaths the iterative version configured. As the number of available wavelength channels increases from $W = 4$ to 14 the both algorithms manage to configure more lightpaths starting from Copenhagen. The connections the algorithm configures appear to be logical.

In fig. 4 the volume of carried traffic on single hop is depicted for each algorithm as a function of available wavelength channels. On y-axis is the volume of single hop traffic, i.e. amount of traffic that reaches its destination on single hop, and on x-axis is the number of available wavelength channels. From the figure it can be noted that both *CP1e* and the iterative version of *CP1* find a configuration which carries all the traffic in single hop (full connectivity) with several wavelength channels earlier than the simple *CP1* heuristics ($W = 10$ vs. $W = 14$) do. This is partly due to the fact that the other two versions are allowed to use alternative routes, but also the order in which the lightpaths are configured plays an important role.

Finally, fig. 5 shows the number of lightpaths each algorithm configures in the network as a function of wavelength channels available. It can be noted that *CP1e* is able to configure more lightpaths than the other algorithms, especially when the network resources are scarce.

5 Conclusions

In this paper we have studied logical topology design (LTD) using greedy heuristic algorithms to maximize the single hop traffic. The MILP formulation of LTD leads to computationally intractable problems for any network of reasonable size and leaves heuristic algorithms as the only possible practical solution. On the other hand, the maximization of the single hop traffic resembles closely the objectives set by logical topology design problem and seems to be a good alternative design objective.

In [8] a simple and robust algorithm, *CP1*, was presented for the maximization of the single hop traffic. In this paper we have presented two improved versions of *CP1*, namely *CP1i* (iterative) and *CP1e* (enhanced). The iterative version combines *CP1* with any static RWA algorithm, while *CP1e* incorporates new heuristic rules which improve the performance of the algorithm when compared to basic *CP1*. By means of numerical simulations both versions were shown to improve the performance.

Furthermore, a layered RWA algorithm to solve a static RWA problem was presented in appendix A. A numerical example suggests that the layered RWA algorithm performs fairly well when compared to other greedy heuristics. As a fast and robust algorithm with reasonably good performance, the layered RWA algorithm is a good candidate to be incorporated into the iterative *CP1*.

References

[1] T. E. Stern and K. Bala, *Multiwavelength Optical Networks: a Layered Approach*. Addison Wesley, 1999.

[2] R. Ramaswami and K. N. Sivarajan, *Optical Networks, A Practical Perspective*. Morgan Kaufmann Series in Networking, Morgan Kaufmann Publishers, 1998.

[3] R. Ramaswami and K. N. Sivarajan, Design of logical topologies for wavelength-routed optical networks, *IEEE J. Select. Areas Commun.*, vol. 14, pp. 840–851, June 1996.

[4] R. Dutta and G. N. Rouskas, A survey of virtual topology design algorithms for wavelength routed optical networks, Tech. Rep. TR-99-06, Department of Computer Science, North Carolina State University, 12, 1999.

[5] E. Leonardi, M. Mellia, and M. A. Marsan, Algorithms for the topology design in WDM all-optical networks, *Optical Networks Magazine "Premiere Issue"*, vol. 1, pp. 35–46, Jan. 2000.

[6] M. Mellia, A. Nucci, A. Grosso, E. Leonardi, and M. A. Marsan, Optimal design of logical topologies in wavelength-routed optical networks with multicast traffic, *Globecom 2001*, (San Antonio, Texas), Nov. 2001.

[7] M. A. Marsan, A. Grosso, E. Leonardi, M. Mellia, and A. Nucci, Design of logical topologies in wavelength-routed IP networks, *Photonic Network Communications*, vol. 4, pp. 423–442, Dec. 2002.

[8] Z. Zhang and A. S. Acampora, A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength reuse, *IEEE/ACM Trans. Networking*, vol. 3, pp. 281–288, June 1995.

[9] D. Banaree and B. Mukherjee, Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study, *IEEE/ACM Trans. Networking*, vol. 8, pp. 598–607, Oct. 2000.

[10] S. Baroni, *Routing and wavelength allocation in WDM optical networks*. PhD thesis, University College London, May 1998.

A Static RWA

The (static) routing and wavelength assignment (RWA) is a subproblem of LTD problem where the aim is to configure a given logical topology onto the given physical topology, or in other words, to establish a given set of lightpaths in the optical layer. The physical constraints are typically the maximum number of wavelength channels, W, available and the distinct channel assignment (DCA) constraint. The logical node degree, i.e. the number of transceivers in each node, should have already be taken into account when the set of lightpaths to be established was defined.

The algorithm described next tries to solve the static RWA problem so that the chosen routing takes into account the restrictions from the wavelength assignment, while still being very fast. The algorithm resembles closely the *CP1* algorithm 1 originally proposed in [8]. The main difference is that algorithm 1 tries to configure as many lightpaths as possible in the order defined by the single hop traffic they would carry, while this algorithm tries to configure a certain set of lightpaths using any feasible routes. The layered RWA algorithm is described formally in algorithm 4.

The performance of the presented algorithm is limited by the set of paths defined by the parameter S and, especially, by their ordering. In fact the optimal configuration, in the sense of the number of channels needed, can always be reached by the layered RWA algorithm: let Z_i be the set of paths at layer i in the optimal configuration and arrange $\mathcal{X}$ so that paths in the beginning are paths in Z_0, then paths in Z_1 etc., which clearly leads to the optimal solution. In practice the optimal configuration is not known, but a reasonably good ordering is as follows. For each path p we define a vector, $\big(\Delta(p),\ \ -n(p),\ \ p_1,\ \ p_2,\ \ \ldots,\ \ p_{n(p)}\big)$, where $n(p)$ is the path length in hops, $\Delta(p)$ is the number of additional hops when compared to the shortest path, and p_i's are the node numbers. The order of two paths is defined by the first different element in the respective vectors (smaller first).

Algorithm 4 Layered RWA

store in ordered list $\mathcal{X}$ the S shortest paths for all requests
set $W \leftarrow 0$
while $\mathcal{X} \neq \emptyset$ **do**
  set $W \leftarrow W + 1$
  for each $(s, d, p) \in \mathcal{X}$ **do**
    if p fits in layer W **then**
      assign path p at layer W for lightpath (s, d)
      remove all paths $s \rightarrow d$ from $\mathcal{X}$
    end if
  end for
end while
return W

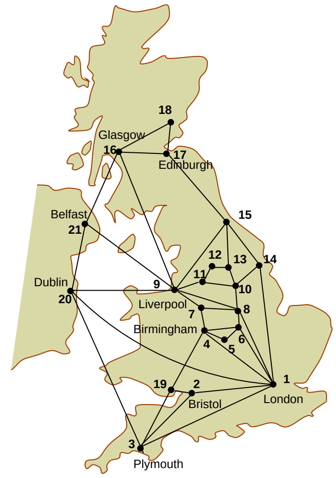

Figure 6: UKNet, a telephone network located in UK consists of 21 nodes and 39 links.

The network illustrated in fig. 6 is used to validate the performance of the layered RWA algorithm. The logical topology to be established consists of lightpaths between every (s, d) pair. As a reference we use a greedy algorithm, which first fixes one shortest path for each lightpath and then assigns wavelength channels using a greedy node coloring algorithm. Table 2 contains the numerical results obtained with different heuristic algorithms. The "bidirectional" corresponds to the situation where each lightpath is bidirectional, i.e. the same path is used in both directions between each node pair. Similarly, the "unidirectional" corresponds to the case where a lightpath $a \rightarrow b$ can traverse a different route than the lightpath $b \rightarrow a$.

Algorithm	unidirectional	bidirectional
Greedy	32	31
Layered	22	23
Baroni [10]	-	20

Table 2: Results of fully-connected UKNet.

The results suggest that layered RWA algorithm performs very well. The result from [10] is obtained with a considerably more complex algorithm and is presented here only for comparison purposes.