Maximization of Single Hop Traffic with Greedy Heuristics

Esa Hyytiä*
Networking Laboratory, HUT

20th December 2002

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 greedy heuristics configure a lightpath at a time onto the network. In this paper two new variations of the previously proposed heuristic algorithm are presented and evaluated. Especially, it is shown by numerical examples that by using an appropriately chosen order on how the lightpaths are configured onto the network it is possible to find reasonably good configurations which do not waste the network resources unnecessarily.

Keywords: WDM, RWA, logical topology design, single hop traffic

1 Introduction

The wavelength division multiplexing (WDM) is a way to exploit efficiently the vast capacity of the optical fibres. In the all-optical wavelength routed (WR) networks the routing in the optical layer is based on wavelength channels offering transparent optical pipes through the network [SB99, RS98]. These characteristics make them 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. In figure 1 the possible solutions are depicted. Eventually the trend is towards IP-over-Optical solutions, where IP packets are transferred directly in the optical layer without any intermediate layer. For example in IETF a work in 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.

*email: esa@netlab.hut.fi

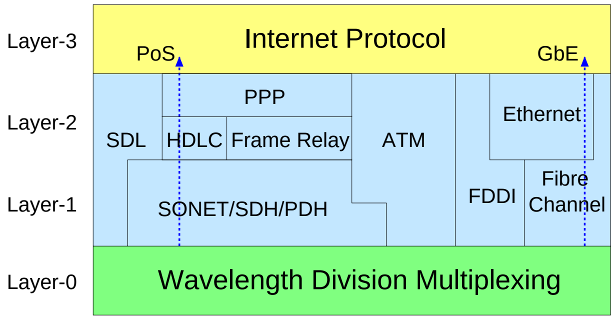

Figure 1: Different approaches for IP-over-Optical (WDM).

A wavelength routed 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, i.e. routing for both the optical and the logical layer. During the network optimization a tradeoff must be made between the huge capacity in the fibre with the electro-optic conversion and electronic processing time in the logical layer.

The topic has been studied a lot and in the literature one can find several formulations for the LTD problem, see e.g. [RS96, DR99, LMM00, MNG$^+$01, MGL$^+$02]. 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 [ZA95] the authors proposed to maximize the single hop traffic as an alternative optimization goal. This simplifies the 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, called *CP1*, to solve the single hop maximization problem. The heuristic configures one lightpath at a time between such nodes where the volume of single hop increases the most. In this paper we propose two alternative greedy heuristics. The first heuristic, 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, denoted with *CP1e*, is similar to *CP1* but uses a novel dynamic order in which it configures the lightpaths. Both algorithms proposed 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 proposed in [ZA95] 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 will 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 [RS96, DR99, BM00], are presented in table 1. The main difference is that we allow any number of physical fibres between each node pair, as well as, any number of lightpaths between each node pair. With these definitions the optimization problem can be formulated as follows:

Objective function: 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:

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

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

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

Traffic Routing: (logical layer)

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

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

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

(flow conservation)

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

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

Routing and Wavelength Assignment: (optical layer)

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

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

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

(lightpath continuity)

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

constant	explanation
p_{ij}	**number of physical fibres** $i \to j$, 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 $i \to j$ can traverse (constraint).
$\Delta_{\max}^{(\text{in})}$	**maximum logical in degree**, i.e. number of optical receivers.
$\Delta_{\max}^{(\text{out})}$	**maximum logical out degree**, i.e. number of optical transmitters.
$\lambda_{\max}$	**maximum congestion** in the logical layer, $\lambda_{\max} \leq \max_{i,j} \lambda_{ij}$.
variable	**explanation**
b_{ij}	**number of lightpaths** $i \to j$.
$c_{ij}^{(k)}$	**number of lightpaths** $i \to j$ **using the wavelength channel** k.
$c_{ij}^{(k)}(l,m)$	**number of lightpaths** $i \to j$ **using the wavelength channel** k **at link** $l \to m$.
$\lambda_{ij}^{(sd)}$	**proportion of traffic** from s to d routed through lightpath $i \to j$.
λ_{ij}	**virtual traffic load** in lightpath $i \to j$, consists of proportions $\lambda_{ij}^{(sd)}$.

Table 1: Notation.

(hop constraint)
$$\sum_{lm} c_{ij}^{(k)}(l,m) \leq h_{ij}, \qquad \forall\,(i,j),\ k \tag{2m}$$

(value range)
$$c_{ij}^{(k)} \in \{0,1\}, \qquad \forall\,(i,j),\ k \tag{2n}$$
$$c_{ij}^{(k)}(l,m) \in \{0,1\}, \qquad \forall\,(i,j),\ (l,m),\ k \tag{2o}$$

Note that the traffic in the logical layer can be bifurcated to any number of flows. In practice it is very unlikely that the network operators will allow this but instead use one (or few) route(s) for all the traffic between each $s-d$ pair. Requiring this, however, makes the traffic routing subproblem even harder to solve. Thus such requirement is not included nor considered in this paper.

2.1 Alternative Objective Functions

The presented MILP formulation minimizes the average number of hops a packet takes [BM00], i.e. how many times a packet needs to be processed on average in the electronical layer before it reaches its destination. Note that this is a linear function, as the factor in front of the summation is a constant for a given traffic matrix.

Alternatively one could minimize the maximum congestion $\lambda_{\max}$. In [RS96] the authors limited themselves to the case of a single fibre per link and at most one lightpath between

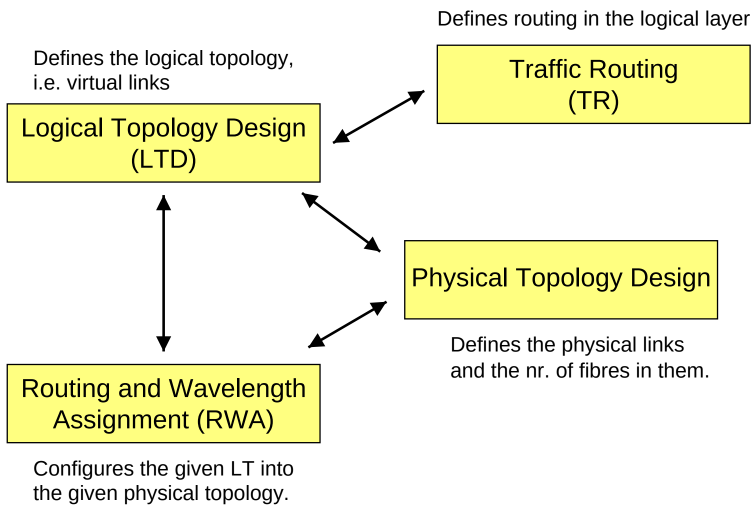

Figure 2: Relationships between the different subproblems. The global optimum requires taking each subproblem into account.

any node pair. With these restrictions the (almost) identical formulation turns out to be a MILP problem [RS96].

Also in [PKG02] the authors have proposed to primarily minimize the maximum congestion $\lambda_{\max}$. As this objective function typically leaves a plenty of freedom on how to configure non-congested links/lightpaths the authors propose to minimize the average packet hop distance as a secondary objective once the minimal $\lambda_{\max}$ has been obtained. The problem that the presented MILP formulation allows the traffic to bifurcate onto numerous lightpaths in arbitrary proportions is also addressed in [PKG02].

Furthermore, in the designing phase one could ask how many electronical interfaces are needed to support a given traffic matrix, i.e. the number of electronical interfaces would be the objective function to be minimized.

2.2 Decomposition of LTD

The MILP formulation presented is often intractable and one must try other sub optimal approaches to tackle with the problem. A straightforward approach is to solve the joint LTD problem in parts. Namely, we can decompose the joint problem into the following subproblems (see figure 2):

i) Logical topology design (LTD), i.e. Eqs. (2a)-(2c).

ii) Traffic routing (TR), i.e. Eqs. (2d)-(2h).

iii) Lightpath routing and wavelength assignment (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.

Fixing the logical topology (LT), i.e. fixing the lightpaths, is in a way the hardest step to take as one does not know the final value of the objective function until the other subproblems, traffic routing and lightpath establishment, are solved. Nonetheless, one is suppose to decide on the logical topology before these later steps are taken based on the available information. For the logical topology we have two requirements. Firstly, the logical topology must be realizable, that is one must be able to solve the resulting RWA problem without violating the physical constraints (e.g. not exceeding the number of wavelengths available). Secondly, the resulting logical topology should allow an efficient routing at logical layer (e.g. IP layer).

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 [ZA95]. 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,

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

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 [ZA95], 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.

3.1 Greedy Heuristics for Single Hop Maximization

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]

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

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

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,

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

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

As was mentioned earlier, in [ZA95] maximization of the single hop traffic was first proposed. First the authors formulated the 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. Thus the authors in [ZA95] also suggested an approach where lightpaths are established on one wavelength layer at a time. The problem of maximizing single hop traffic in one wavelength layer was named as *CP1*, and it is clearly equivalent to *CP* with one wavelength layer. 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.

3.2 *CP1* Heuristic

The authors in [ZA95] 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.

3.3 Iterative *CP1*

The *CP1* heuristic 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 section A.2. If the used static RWA algorithm manages to find more "space", then the *CP1* heuristic 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. It is clear that as the iterative version only configures additional lightpaths to the solution(s) of the basic *CP1*, the

Algorithm 1 Heuristic Algorithm for *CP1* [ZA95]

let $\mathcal{N}$ be the set of network nodes
set $t_{s,d} \leftarrow \lambda^{(sd)}$ {the initial traffic intensity from node s to node d}
$\mathcal{X} \leftarrow$ (one of) the shortest path ℓ for each node pair $(s,d) \in \mathcal{N} \times \mathcal{N}$
$W \leftarrow 0$
while $W < W_{\max}$ **do**
  $W \leftarrow W + 1$
  sort the paths in $\mathcal{X}$ 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

amount of single hop traffic will never be less than what *CP1* alone 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.

Algorithm 2 Iterative *CP1*

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

3.4 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 a bit different and reminds closely the ideas behind the layered RWA algorithm 6. Furthermore, 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).

Thus, 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. The order is clearly well-defined.

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 $z(p)$ and remove it from $\mathcal{X}'$
    if path p is free at layer W **then**
      Configure a lightpath p at layer W
      Update traffic demands: $t_{i,j} \leftarrow \max\{0, t_{i,j} - 1\}$
    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

The fact that algorithm 3 configures the longer paths (which need more resources) 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 of the algorithm.

Note that for each layer the order is dynamic unlike the case was with *CP1*. Furthermore, as the order reminds closely the order of the layered RWA algorithm 6 the iterative version using the layered RWA algorithm to reconfigure the current LT is unlikely to give any improvement. However, using a more sophisticated RWA algorithm may turn out to be successful.

3.5 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 the following approaches have been suggested to guarantee a connected logical topology:

i) In [ZA95] it is proposed that before filling the last wavelength layer one makes sure that the graph is connected by adding necessary lightpaths. Then the algorithm proceeds normally and fills the last wavelength layer normally.

ii) In [RS96] and [BM00] it is proposed that if the (maximum) logical degree is greater than the physical degree then one initially assigns a lightpath to each physical link before configuring any other lightpaths. This clearly ensures a fully connected logical topology. If the traffic is highly localized this should not hinder the solution much. Note that if the heuristic algorithm solving LT includes RWA, as is the case, e.g. with [RS96], one should not assign a wavelength for these short lightpaths yet but instead just reserve one channel, and fix the wavelength when the number of free channels in corresponding link becomes one.

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 figure 3 together with the used traffic matrix (scaled to the capacity of wavelength channel).

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

Figure 3: Cost 239 project test core network.

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 the routing at the logical layer was not determined, but instead the traffic carried with single hop was used as a goodness criterion.

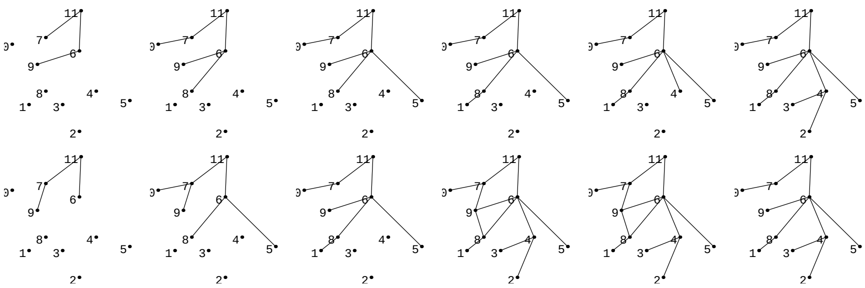

Figure 4: Configured lightpaths from Copenhagen (node 11). Upper row corresponds to the algorithm *CP1* with $W = 4, 6, 8, 10, 12$ and 14 wavelength channels. Lower row corresponds to the iterative version.

In figure 4 the configured lightpaths from Copenhagen are depicted for *CP1* and its iterative version as a function of available wavelength channels. The connections the algorithm configures seem reasonable.

In figure 5 the carried traffic is depicted for each algorithm as a function of available wavelength channels. From the figure it can be noted that both the *CP1e* and 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* heuristic ($W = 10$ vs. $W = 14$) does. 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, figure 6 shows the number of lightpaths each algorithm configures in the network

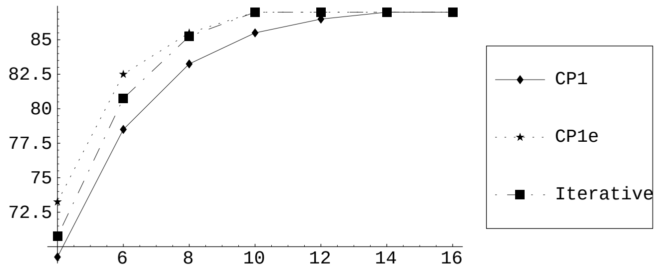

Figure 5: The volume of single hop traffic for *CP1* and an iterative version. On the x-axis is the number of wavelength channels and y-axis represents the volume of single hop traffic.

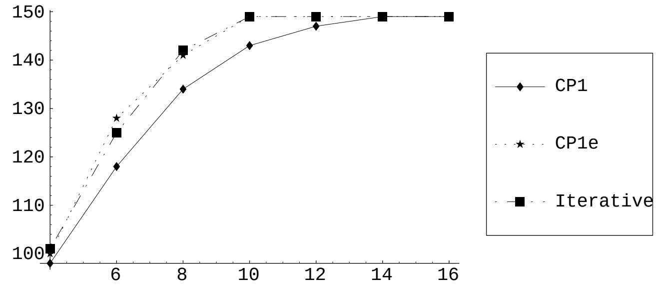

Figure 6: The number of lightpaths (logical links) configured as a function of the number of wavelength channels available.

as a function of wavelength channels available. It can be noted that the *CP1e* configures more lightpaths that the others 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 reasonable size network 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.

In [ZA95] a simple and robust algorithm, *CP1*, is 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*. Both new versions were shown to have a better performance by means of numerical simulations.

Furthermore, a layered RWA algorithm to solve a static RWA problem is presented in section A.2. An example case suggests that the layered RWA algorithm performs fairly well when compared to other greedy heuristics. As being fast and robust algorithm with reasonably good performance, the layered RWA algorithm is a good candidate to be incorporated into the iterative *CP1*.

References

[Bar98] Stefano Baroni. *Routing and wavelength allocation in WDM optical networks*. PhD thesis, University College London, May 1998.

[BG92] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice-Hall, 2nd edition, 1992.

[BM96] Dhritiman Banaree and Biswanath Mukherjee. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *IEEE J. Select. Areas Commun.*, 14(5):903–908, June 1996.

[BM00] Dhritiman Banaree and Biswanath Mukherjee. Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Trans. Networking*, 8(5):598–607, October 2000.

[Bré79] Daniel Brélaz. New methods to colour the vertices of a graph. *Communications of the ACM*, 22(4), April 1979.

[CMLF00] Tibor Cinkler, Daniel Marx, Claus Popp Larsen, and Daniel Fogaras. Heuristic algorithms for joint configuration of the optical and electrical layer in multi-hop wavelength routing networks. In *INFOCOM (2)*, pages 1000–1009, 2000.

[DR99] Rudra Dutta and George N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. Technical Report TR-99-06, Department of Computer Science, North Carolina State University, 12, 1999.

[LMM00] Emilio Leonardi, Marco Mellia, and Marco Ajmone Marsan. Algorithms for the topology design in wdm all-optical networks. *Optical Networks Magazine "Premiere Issue"*, 1(1):35–46, January 2000.

[LS00a] Guangzhi Li and Rahul Simha. The partition coloring problem and its application to wavelength routing and assignment. In *Optical Networks Workshop*, Richardson, Texas, February 2000.

[LS00b] Gang Liu and Thomas E. Stern. An algorithm for optimal routing and wavelength assignment on a bidirectional ring. In *Optical Networks Workshop*, Richardson, Texas, February 2000.

[MGL+02] Marco Ajmone Marsan, Andrea Grosso, Emilio Leonardi, Marco Mellia, and Antonio Nucci. Design of logical topologies in wavelength-routed ip networks. *Photonic Network Communications*, 4(3/4):423–442, December 2002.

[MNG+01] Marco Mellia, Antonio Nucci, Andrea Grosso, Emilio Leonardi, and Marco Ajmone Marsan. Optimal design of logical topologies in wavelength-routed optical networks with multicast traffic. In *Globecom 2001*, San Antonio, Texas, November 2001.

[OBC+00] Teunis Ott, Tony Bogovic, Tami Carpenter, K. R. Krishnan, and David Shallcross. Algorithms for flow allocation for multi protocol label switching, August 2000.

[PKG02] Nicolar Puech, Josué Kuri, and Maurice Gagnaire. Models for the logical topology design problem. In *Networking 2002*, pages 301–313, Pisa, Italy, May 2002.

[RS96] Rajiv Ramaswami and Kumar N. Sivarajan. Design of logical topologies for wavelength-routed optical networks. *IEEE J. Select. Areas Commun.*, 14(5):840–851, June 1996.

[RS98] Rajiv Ramaswami and Kumar N. Sivarajan. *Optical Networks, A Practical Perspective*. Morgan Kaufmann Series in Networking. Morgan Kaufmann Publishers, 1998.

[SB99] Thomas E. Stern and Krishna Bala. *Multiwavelength Optical Networks: a Layered Approach*. Addison Wesley, 1999.

[SBD+00] A. Sridharan, S. Bhattacharyya, C. Diot, R. Guerin, J. Jetcheva, and N. Taft. On the impact of aggregation on the performance of traffic aware routing. Technical report, University of Pennsylvania, July 2000.

[ZA95] Zhensheng Zhang and Anthony S. Acampora. A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength re-use. *IEEE/ACM Trans. Networking*, 3(3):281–288, June 1995.

A Static Routing and Wavelength Assignment

As was mentioned earlier 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 just the maximum number of wavelength channels, W, available and the requirement of the distinct channel assignment (DCA). The logical node degree, i.e. the number of transceivers in each node, should already be taken into account when defining the set of lightpaths to be established.

A.1 Greedy RWA Algorithm

In this section we present a simple greedy algorithm to obtain a reasonable configuration. Algorithm 4 contains a version of Dijkstra's shortest path method which obtains S shortest routes from node s to node d.

Note that the *idx* function described in the algorithm is used to sort the paths in a certain order. The primary key is the number of additional hops, i.e. the paths using additional hops are given a lower priority. The second criterion is the length of the path, the longer paths come before the shorter ones. If these two criteria are the same, then the order is defined by the first different node along the paths; smaller node number comes first.[2] The

[2] when searching for the S shortest paths we use the same logic in case of equally long paths.

Algorithm 4 S shortest paths

1: let $\mathcal{L} = \{(s_i, d_i)\}$ be a list of node pairs for which a lightpath is suppose to be configured
2: find the S shortest paths for each $(s_i, d_i) \in \mathcal{L}$, and store the triples (s_i, d_i, p_i), where p_i is the corresponding path, to a list $\mathcal{X}$
3: let len(p_i) be the length of path p_i (in number of hops)
4: let sp$(p_i) = $ sp(s_i, d_i) be the length of the shortest possible path from s_i to d_i
5: let idx$(p_i) = (N+1) \cdot [\text{len}(p_i) - \text{sp}(p_i)] - \text{len}(p_i) + \sum_{j=1}^{\text{len}(p_i)} p_i(j)(N+1)^{-j}$, where $p_i(j)$ is the node number of jth node in path p_i
6: sort the list $\mathcal{X} = \{(s_i, d_i, p_i)\}$ in increasing order of idx(p_i)
7: return the list $\mathcal{X}$

last criterion is only needed to ensure an unambiguous ordering, which makes it possible to reproduce the simulation results.

Algorithm 5 Greedy RWA

1: use algorithm 4 to find shortest paths for each lightpath request
2: form WA graph $\mathcal{G}$ where each node represents one lightpath, and lightpaths sharing a link are set as neighbours
3: use greedy node coloring algorithm (see e.g. [Bré79]) to color the nodes of $\mathcal{G}$
4: return paths and chosen wavelengths

Algorithm 5 solves the RWA problem in two parts. First it uses the previously described shortest path algorithm to obtain one route for each lightpath request, and then assigns wavelength channels to them using a well-known greedy node coloring algorithm, described, e.g. in [Bré79]. It may also be possible to use more sophisticated graph node coloring algorithms like tabu search or simulated annealing, or even to do an exhaustive search if the size of the problem permits it. Note that the greedy algorithm uses only one shortest path for each lightpath.

A.2 Layered Approach for Static RWA

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 presented in [ZA95]. The main difference is that algorithm 1 tries to configure as many lightpaths as possible in the order of single hop traffic they would carry using a fixed route, while this algorithm simply tries to configure a certain set of lightpaths using any feasible routes.

Similarly as in [ZA95] we split the problem into several subproblems in which our goal is to use the resources of one wavelength layer maximally. First we take wavelength layer 1 and configure as many lightpaths there as possible. After that layer 1 is considered "frozen" and we are left with a new smaller problem where the lightpaths configured to layer 1 are removed from the demand list. By doing this we hope to find a good configuration for one layer at a time leading to a satisfactory overall configuration.

The layered RWA algorithm is described formally in algorithm 6. The algorithm resem-

Algorithm 6 Layered RWA

find S shortest paths using algorithm 4 and store them in ordered list $\mathcal{X}$
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

bles closely the greedy node coloring algorithms. The distinction is that here we fix both the route (node) and color, and only one node corresponding to each $s - d$ pair is given a color [LS00a].

The algorithm can be easily extended to the case where there is more than one lightpath request for some $s-d$ pairs. Namely instead of immediately removing all the paths $s \rightarrow d$ after assigning one path, one decreases “multiplicity” counter for $s \rightarrow d$. Then the paths $s \rightarrow d$ are only removed from the working set $\mathcal{X}$ when the corresponding counter reaches zero.

The performance of the above algorithm is limited by the set of paths defined by the parameter S and, especially, by their ordering (see algorithm 4). Namely, there exists always a constant $S < N$ and some ordering which lead to the global minimum. Thus we have the following proposition.

Proposition 1
Optimal configuration in the sense of the number of channels needed can always be reached by the layered RWA algorithm for some order of paths.

Proof:
Let Z_i be the set of paths at layer i in the optimal configuration. Clearly any ordered set beginning with $\{Z_0\ Z_1\ \ldots\ Z_W\}$ leads to the optimal solution.

The algorithm could possibly be improved by incorporating a more intelligent algorithm to fill one wavelength layer. The used greedy algorithm can assign first a long route which blocks some of the remaining shorter connections. Especially when assigning the last few layers more intelligent choices of routes might turn out advantageous. Also some kind of backtracking algorithm would probably further improve the performance.

A.3 Numerical Results with RWA Heuristics

In order to validate the performance of the presented heuristic RWA algorithms we use the network illustrated in figure 7. The logical topology to be established consists of a

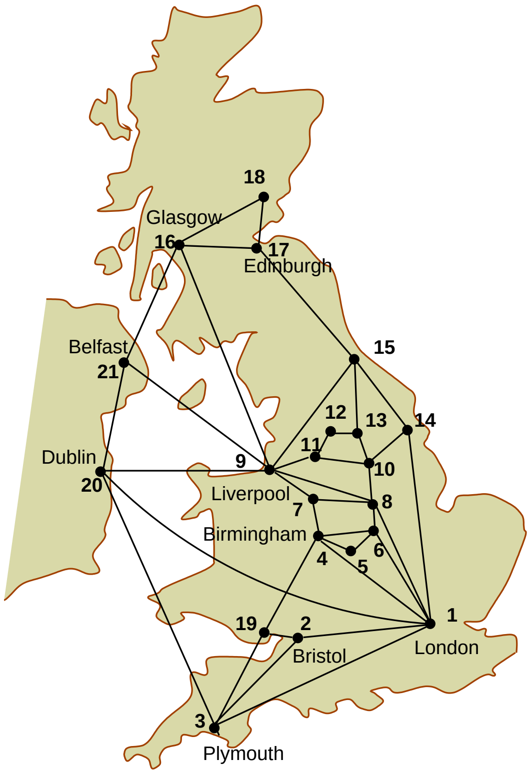

Figure 7: UKNet, a telephone network located in UK consists of 21 nodes and 39 links.

lightpath between every node pair. 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 $a \to b$ can traverse a different route than the lightpath $b \to a$. Clearly the bidirectionality is an additional constraint and thus the optimal unidirectional solutions should never be worse.

Algorithm	unidirectional	bidirectional
Greedy	32	31
Layered	22	23
Baroni [Bar98]	-	20

Table 2: Results of fully-connected UKNet.

It can be noted from the results that the simple greedy heuristic with the shortest path routing does not perform very well. On the other hand, the layered RWA algorithm gives reasonably good results. The results from [Bar98] are for comparison purposes and they were obtained with moderately complex heuristic algorithm. As a conclusion we expect that the layered RWA algorithm, while being very fast, still obtains very good configurations.