



**HAL**  
open science

## Scalable Routing, Scheduling and Virtualization for TWIN Optical Burst Switching Networks

Ion Popescu, Annie Gravey, Bogdan Uscumlic, Yvan Pointurier, Ahmed Triki,  
Philippe Gravey

► **To cite this version:**

Ion Popescu, Annie Gravey, Bogdan Uscumlic, Yvan Pointurier, Ahmed Triki, et al.. Scalable Routing, Scheduling and Virtualization for TWIN Optical Burst Switching Networks. NOC 2015: 20th European Conference on Network and Optical Communications, Jun 2015, Londres, United Kingdom. pp.1-6, 10.1109/NOC.2015.7238609 . hal-01214048

**HAL Id: hal-01214048**

**<https://hal.science/hal-01214048>**

Submitted on 30 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# Scalable Routing, Scheduling and Virtualization for TWIN Optical Burst Switching Networks

Ion Popescu\*, Bogdan Ušćumlić†, Ahmed Triki\*, Yvan Pointurier†, Annie Gravey\* and Philippe Gravey\*

\*Institut Mines Télécom, Télécom Bretagne, Brest, France

(email: *firstname.lastname@telecom-bretagne.eu*)

†Alcatel-Lucent Bell Labs, Nozay, France

(email: *firstname.lastname@alcatel-lucent.com*)

**Abstract**—We propose a comprehensive scalable algorithm simultaneously assessing the routing, scheduling and virtualization in Time-Domain Wavelength Interleaved Network (TWIN). TWIN is an energy-efficient optical burst switching technology for metropolitan area and data center intra- and inter-connections, with destinations using separate allocated sets of wavelengths for the reception. Given the costs of the optical transponders and the wavelength use per km of fiber length, the proposed algorithm solves the routing and wavelength assignment problem, and in the same time allocates the time slots to traffic flows (performs the slot scheduling), so that the total network cost is minimized. The algorithm also enables the construction of overlaid virtual networking domains at minimum cost, which share the transponders and network links, and can employ different scheduling policies. The performance of the algorithm is compared with the pre-existing optimal dimensioning solution for single virtualization domain based on Integer Linear Programming, for different scenarios. The obtained results show that the network cost is within 27% of the optimal.

**Index Terms**—Optical burst switching, scheduling, network virtualization, routing and wavelength assignment.

## I. INTRODUCTION

The increase of bandwidth demand in the metropolitan networks and the scalability limitations of Ethernet based data centers, foster the need for the new optical transport layer that shall ensure a highly efficient bandwidth utilization of the optical resources, by, in the same time, improving the network energy efficiency. This paper deals with Time-Domain Wavelength Interleaved Network (TWIN), an optical burst switching transport technology, proposed in [1], that is candidate solution both for metropolitan and all-optical data center inter- and intra-connection networks [2]. In order to achieve a high energy efficiency, all the complexity in TWIN network is pushed to the network edge. Edge nodes are complex and perform the main functions, including all those related to data, management and control plane. These nodes take the roles of sources and/or destinations (see Fig. 1). Inside of the network are the passive optical components that are pre-configured and passively perform the switching of the arriving optical bursts according to wavelength of their destinations. Such network configuration is possible thanks to the wavelength based destination addressing, where a single wavelength is allocated to each edge node in a TWIN network.

Because of the specific wavelength distribution to the nodes, TWIN can be seen as an overlap of multi-point to point

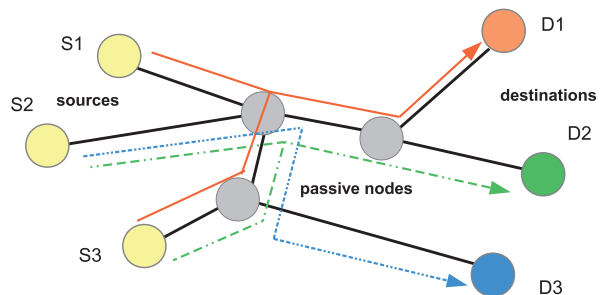


Fig. 1. TWIN architecture. All complexity is implemented in the edge nodes S1, S2, S3, D1, D2, D3, while the other nodes (“passive nodes”) only require simple optically passive components.

lightpath trees, each rooted at a fixed-wavelength (colored) receiver on a (destination) node. The operation of the network is usually slotted (as considered here), and the slots can be either of fixed or variable size. Scheduling of time slots to the traffic flows is the key issue in TWIN, as it impacts the overall network performance and the network capacity use. TWIN uses transponders composed of fixed-wavelength receiving part and fast-wavelength tunable transmitting part.

The scheduling in TWIN has been extensively studied by different research groups, with accent on minimizing the schedule length, centralized and distributed scheduling, and different traffic dynamicity (e.g., see [3],[4],[5]). The work in [6], for the first time, simultaneously addresses the Routing and Wavelength Assignment (RWA) and scheduling problem in TWIN, and argues that these two problems shall be solved together to minimize the total network cost. The proposed dimensioning tool is an Integer Linear Program (ILP).

Optimally solving the joint RWA and scheduling problem in TWIN cannot be done in polynomial time, so the linear programming dimensioning tools proposed in previous works are limited to small examples and cannot be used for real-scale networks. In this paper, we propose a heuristic method, called Heuristic Method for Virtual TWIN Dimensioning (HMVTD), that as primary concern provides a scalable solution for the RWA and scheduling optimization, a problem treated in [6]. In addition, for the first time, the proposed algorithm allows the creation of the arbitrary number of virtualization domains over the same physical TWIN network, at minimum cost, and thus assesses the emerging challenge of enabling the network level virtualization for optical transport technologies [7],[8],[9].

The network virtualization is an important building block in Software-Defined Networking (SDN) [7], and is a technology that shall facilitate the centralized control and the automation of the high capacity optical infrastructure. The optical network virtualization enables the abstraction and sharing of optical interfaces, channels and links between different virtual network domains, and offers to the service providers the possibility to disjointly operate such domains over the same optical infrastructure. Separate virtualization domains usually support separate QoS guarantees and service contracts (e.g., see [10]). To the best of the authors' knowledge, a virtualization enabling algorithm has not been proposed so far for a TWIN network. HMVTD defines a way the optical interfaces are shared between different virtualization domains, and maps the time slots to each of the interfaces, according to the calculated global scheduling.

The remainder of the paper is organized as follows. Section II describes the proposed heuristic method and its variants. Section III presents the detailed numerical results. Finally, Section IV concludes the paper and highlights the contributions of this work.

## II. HEURISTIC METHOD FOR VIRTUAL TWIN DIMENSIONING

The CAPEX cost of the TWIN network, or the total network cost, accounts for the wavelength leasing cost per km of the used fiber, ( $C_{lw}$ ), and the cost of each transponder, TRX ( $C_t$ ). The objective of the dimensioning problem considered here is to minimize the total network cost, over the all network virtualization domains. We use the traffic matrix to identify the number of needed slots for each traffic demand. Both the size of the TWIN's schedule  $K$  and the input traffic matrix are the input parameters for the heuristic. We assume that the propagation delays are integer number of slot durations. Finally, as in [6], we suppose that the slots that are the object of allocation are enumerated by numbers  $0, 1, \dots, K-1 \pmod{K}$ , and that the scheduling being the result of the optimization is periodically repeated. The full list of input parameters and variables used by HMVTD algorithm is given next.

### A. Input Parameters and Variables

#### Input Parameters

- $G(V, E)$ : a non-directed graph describing the physical mesh topology, where  $V$  is the set of nodes and  $E$  is the set of links;
- $T_{i,j}^v$ : traffic demand in number of slots (normalized to the schedule length) to be allocated to a flow between the source  $i$  and destination  $j$ , for virtual domain  $v$ ;  $\ell_{i,j}$ : length in km of a link connecting the node  $i$  to node  $j$ ;  $D_k$ : time slot duration;  $\nu$ : speed of light in the fiber;  $W$ : maximum allowed number of wavelengths per fiber;  $TRX$ : maximum allowed number of transponders per node;  $C_{lw}$ : wavelength leasing cost per km of the used fiber;  $C_t$ : transponder cost;  $K$ : number of slots used for the allocation, (i.e. the "schedule length");

- $DemandOrderingPolicy(v)$  (either MLC, MLS, MLD, LCF or RD),  $DemandServingPolicy(v)$  (either ED or PD),  $SlotSelectionPolicy(v)$  (either FFS or RS), with the acronymes defined below; These policies can be different for different virtualization domains  $v$ .
- $N$ : number of iterations (constant  $\geq 1$  if  $DemandOrderingPolicy(v)=RD$  or  $SlotSelectionPolicy(v)=RS$ , else equal to 1);
- $D$ : maximum number of overlaid virtual networking domains in the final solution;
- $NetworkConfiguration$ : contains the mapping of slots and virtual domains to the physical network (it can be a given parameter, or an empty set).

#### Output Variables

- $R_{i,j}^v$ : set of links used to route the demand from the source  $i$  to the destination  $j$ , in the virtual domain  $v$ ;  $L_{i,j}^v$ : length of path from source  $i$  to destination  $j$ , in the virtual domain  $v$ ;  $\delta_{i,j}^v$ : propagation delay (in number of slots) experienced by a slot emitted by the source  $i$  until it reaches the destination  $j$ , for virtual domain  $v$ ;
- binary  $S_{i,j}^{t,w,v}(k)$  equal to 1 if slot  $k$  is used to carry demand from source  $i$  to destination  $j$ , in virtual domain  $v$ , by using transmitter  $t$ , wavelength  $w$ , and equal to 0 otherwise;
- binary  $tx_i^t$  equal to 1 if a transmitter  $t$  with fast tunable laser is deployed at node  $i$ , and 0 otherwise; binary  $rx_j^w$  equal to 1 if a fixed wavelength receiver  $w$  is deployed at node  $j$ , and 0 otherwise;
- integer  $N_W$ , denotes total number of wavelengths used in the network; real  $C_W$  denotes total wavelength leasing cost in the network; real  $C_{TRX}$  denotes total transponder cost in the network; real  $C_{TOT}$  denotes total design cost of the network;

The pseudo-code for the algorithm HMVTD is given in Alg. 1, and is composed of three basic parts: 1) tree and path construction, 2) resource allocation, 3) cost calculation. The algorithm is repeated  $D$  times in order to calculate the overlapping slots and define the optical resource sharing for all virtual domains. The details of all algorithm parts are explained next.

### B. Tree and path construction

In this step, first the minimum spanning tree (MST) is found for a given virtual domain. Several algorithms find the MST, like Kruskal's [11] and Steiner's algorithms. While Kruskal algorithm spans all vertices ( $V$ ) of a given graph, the Steiner algorithm spans a given subset of vertices ( $V' \subset V$ ). Therefore, for the current implementation, the Kruskal's algorithm is chosen.

Next, the routing paths  $R_{i,j}^v$  are found on the MST: the problem is solved by Breadth-First Search (BFS) algorithm. BFS finds the path on the graph starting from the source  $i$  and inspects all the neighbouring nodes until it finds the destination  $j$ . At the end of this step, the delay  $\delta_{i,j}^v$  (in integer number of slots) is calculated.

### C. Resource allocation

For resource allocation, we study various variants of demand ordering policies, demand serving policies, and slot selection policies, as described next.

1) *Demand Ordering Policies*: (a) Most Loaded Connection (*MLC*): the demands  $i, j$  are sorted in the decreasing order based on the load from a source  $i$  towards a destination  $j$ ; (b) Most Loaded Source (*MLS*): the demands  $i, j$  are sorted in the decreasing order based on the total load from a source  $i$ ; (c) Most Loaded Destination (*MLD*): the demands  $i, j$  are sorted in the decreasing order based on the total load towards destination  $j$ ; (d) Longest Connection First (*LCF*): the demands  $i, j$  are sorted in the decreasing order based on the path length  $L_{i,j}^v$  between source  $i$  and destination  $j$ . Note that for the above policies, the ties are broken so that the demand having the lower index is ordered first; (e) Random Demands (*RD*): the demands  $i, j$  are sorted in random order.

2) *Demand Serving Policies*: (a) Entire Demand (*ED*): once a demand is selected, the entire number of required slots  $k$  is allocated; (b) Partial Demand (*PD*): once a demand is selected, a single slot  $k$  out of required number is allocated, and the following demand is served.

3) *Slot Selection Policies*: (a) First Fit Slot (*FFS*): the first available slot  $k$  is allocated; (b) Random slot (*RS*): random available slot  $k$  is allocated.

Note that the slot selection policy can include the QoS specifications (e.g., as done in [6]), which allows to set the desired (separate) QoS level in each virtual domain.

To allocate a slot  $k$  for a demand  $i, j$  on wavelength  $w$ , the slot availability at the source side and destination side is checked. The “NO SLOT BLOCKING” condition from Alg. 1 is defined by the equations:

$$\sum_{j'} \sum_w \sum_v S_{i,j'}^{t,w,v}(k) = 0, \quad \forall k \leq K, \forall t \leq TRX, \forall i \leq |V|; \quad (1)$$

$$\sum_{i'} \sum_t \sum_v S_{i',j}^{t,w,v}(k + [(\delta_{i,j} - \delta_{i',j}) \bmod K]) = 0, \quad \forall k \leq K, \forall w \leq W, \leq K, \forall i \leq |V|, \forall j \leq |V|; \quad (2)$$

Once a slot  $k$  is allocated to the nodes  $i, j$ , the traffic demand  $T_{i,j}^v$  is decreased by 1. The resource allocation sub-problem repeats until all the traffic demands are satisfied.

### D. Cost calculation

The “COST UPDATE” procedure from Alg. 1 consists in updating all the relevant network costs. In this phase, the total design cost  $C_{TOT}$  is calculated, composed from the cost of wavelength use per km of fiber and the cost of transponders.

The total number/cost of transponders  $C_{TRX}$  is calculated (eq. 5) as sum of the maximum between the number of transmitters (eq. 3) and receivers (eq. 4) at each node:

$$D \cdot |V| \cdot W \cdot K \cdot tx_i^t \geq \sum_v \sum_j \sum_w \sum_k S_{i,j}^{t,w,v}(k),$$

**for** all  $v \leq D$  **do**

sub. 1: Tree and path construction:

Run the algorithm MST on  $G(V, E)$ ;

// Now we have a new graph  $G'(V, E')$ ;

Run the algorithm BFS on  $G'(V, E')$ , to calculate  $R_{i,j}^v$  and  $L_{i,j}^v$ , for all  $i, j \in V$ ;

**set**  $\delta_{i,j}^v = L_{i,j}^v / (\nu * D_k)$  ( $\forall i, j \in V$ );

sub. 2: Resource allocation:

Create the *List(Demands)* containing the traffic demands  $T_{i,j}^v$ , sorted in the order defined by *DemandOrderingPolicy(v)*;

**for** iteration  $\leq N$  **do**

**set**  $S_{i,j}^{t,w,v}(k) = 0, (\forall k)(1 \leq k \leq K)$ ;

**for** all

$T_{i,j}^v$ , taken in the order of *List(Demands)* **do**

**while**  $T_{i,j}^v > 0$  **do**

**for** all  $t, w$  taken in the first-fit order **do**

**for** all values of  $k$ , chosen according to *DemandServingPolicy(v)* and *SlotSelectionPolicy(v)* **do**

**if** NO SLOT BLOCKING neither at source nor at destination **then**  
 $S_{i,j}^{t,w,v}(k) = 1; T_{i,j}^v = T_{i,j}^v - 1;$   
 proceed with next  $T_{i,j}^v$ ;

**end**

**end**

**end**

**end**

sub. 3: Cost calculation:

COST UPDATE;

iteration = iteration + 1;

**end**

$v = v + 1$ ; Update *NetworkConfiguration*;

**end**

**Algorithm 1:** Heuristic Method for Virtual TWIN Dimensioning.

$$\forall t \leq TRX, \forall i \leq |V|; \quad (3)$$

$$D \cdot |V| \cdot TRX \cdot K \cdot rx_j^w \geq \sum_v \sum_i \sum_t \sum_k S_{i,j}^{t,w,v}(k),$$

$$\forall w \leq W, \forall j \leq |V|; \quad (4)$$

$$C_{TRX} = C_t \cdot \sum_i \max \left( \sum_t tx_i^t, \sum_w rx_i^w \right); \quad (5)$$

The total number of wavelengths  $N_W$  and the total wavelength cost  $C_W$  per km of fiber are computed as follows:

$$N_W = \sum_j \sum_w rx_j^w; \quad (6)$$

TABLE I  
COST PENALTY FOR DIFFERENT RESOURCE ALLOCATION POLICIES

Demand Ordering Policy	Demand Serving Policy	Slot Selection Policy	Calculation Complexity	Cost Penalty [%]
MLC	ED	FFS	$O(D V ^3K^2)$	68.75
		RS	$O(D V ^2K^3N)$	30.21
	PD	FFS	$O(D V ^3K^2)$	89.58
		RS	$O(D V ^2K^3N)$	46.66
MLS	ED	FFS	$O(D V ^3K^2)$	68.75
		RS	$O(D V ^2K^3N)$	30.16
	PD	FFS	$O(D V ^3K^2)$	89.58
		RS	$O(D V ^2K^3N)$	45.51
MLD	ED	FFS	$O(D V ^3K^2)$	41.66
		RS	$O(D V ^2K^3N)$	29.58
	PD	FFS	$O(D V ^3K^2)$	81.25
		RS	$O(D V ^2K^3N)$	44.37
LCF	ED	FFS	$O(D V ^3K^2)$	39.58
		RS	$O(D V ^2K^3N)$	27.51
	PD	FFS	$O(D V ^3K^2)$	47.91
		RS	$O(D V ^2K^3N)$	44.37
RD	ED	FFS	$O(D V ^3K^2N)$	27.51
		RS	$O(D V ^2K^3N^2)$	29.58
	PD	FFS	$O(D V ^3K^2N)$	44.58
		RS	$O(D V ^2K^3N^2)$	40.62

$$C_W = C_{lw} \cdot \sum_v^D \sum_i^{|V|} \sum_j^{|V|} \sum_w^W (L_{ij}^v \cdot r x_j^w); \quad (7)$$

For the combination which uses the random policies (slot selection policy (RS) and/or demand ordering policy (RD)), the HMVTD computes the total cost  $N$  times (for each virtual domain  $v$ ), and out of  $N$  costs it keeps the value of the minimum total cost,  $C_{TOT}$ .

### III. NUMERICAL RESULTS

This section reports the results obtained by using the HMVTD, which is implemented in C++ programming language. The results are compared with the results obtained from a commercially available LP solver, in which the optimal 0-1 ILP formulation from [6] has been implemented (the latter results are given with the optimality gap of 10%). We consider the 6-node physical topology with 9 links as depicted in Fig. 2. The propagation delay on each link corresponds to 5 slots duration, where one time slot is equal to  $10 \mu\text{s}$ . The wavelength channel and TRX data rate is supposed to be 10 Gb/s. The schedule length is supposed to be  $K = 5$  and the number of wavelengths in the network is limited to  $W = 40$ .

The cost is expressed in arbitrary units (a.u.). We set the TRX cost to  $C_t = 1$  (a.u.), while the cost of wavelength use per km of fiber length (hereafter denoted simply by “wavelength cost”) varies from  $C_{lw} = 10^{-4}$  (a.u.)·km<sup>-1</sup> (lower bound) to  $C_{lw} = 0.1$  (a.u.)·km<sup>-1</sup> (upper bound). The lower bound on  $C_{lw}$  is calculated as a value at which a single transponder cost is equal to the total possible wavelength leasing cost,  $C_t / (\sum_\ell L_\ell \cdot W)$ ; this ensure that no transponder can be traded for a wavelength [6]. The assumption for the upper bound on  $C_{lw}$  is derived from [12], where the ratio

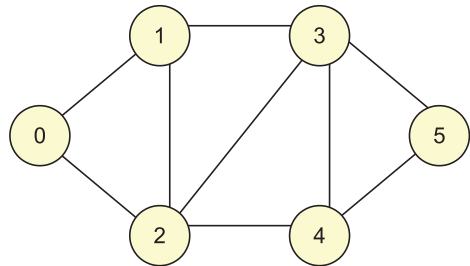


Fig. 2. The logical topology of a 6-node mesh network, with 9 links, used for computer simulations.

between the wavelength cost and the receiver cost is estimated on a small size metropolitan ring.

In the following, we assess the performance of different heuristic combinations in terms of complexity and design cost, first, for the single virtualization domain (in subsections A, B and C). Then, in subsection D, we evaluate the gain of virtualization considering multiple virtual domains.

#### A. Heuristic performance evaluation

We first study the efficiency of the HMVTD using different heuristic combinations, when performing the resource allocation. For this study, we considered a uniform and symmetric traffic between each pair of nodes, with fixed total sent/received traffic of  $\alpha = 2c$  at each node (where  $c$  is the channel capacity). The TRX and wavelength costs are, respectively:  $C_t = 1$  (a.u.),  $C_{lw} = 0.1$  (a.u.) km<sup>-1</sup>. The difference in the cost between the HMVTD and the ILP optimal solution is called “the cost penalty”.

Note that for the heuristics which use randomness in slot selection policy (i.e., RS) and in demand ordering policy (i.e., RD), the number of iterations  $N$  is equal to 100. All the results are averaged over 20 simulations.

Tab. I shows that selecting the demands according to the MLC and MLS achieves worse performance than other algorithm variants. LCF and RD policies have similar cost penalty, while RD policy has slightly increased time complexity due to the presence of iterations.

ED based demand serving policy outperforms the PD in all cases. This proves that serving the demands according to round robin process leads to resource wastage.

RS policy outperforms the FFS in most of the cases. However, FFS achieves low cost penalty (27.51%) when it is coupled with random demand ordering. This is explained by the fact that random process explores multiple possibilities and keeps the configuration with minimum cost.

In addition to the cost penalty, performance of the heuristics is also impacted by their calculation complexity. In HMVTD, the calculation complexity depends on the number of virtual domains ( $D$ ), the network size ( $|V|, |E|$ ), the schedule length ( $K$ ), and the number of iterations ( $N$ ) (an additional parameter for the random-based algorithms). The MST and BFS have the complexity of  $O(|E| \log |V|)$  [11] and  $O(|V| \sqrt{|E|})$  [13] respectively. The complexity of the resource allocation sub-problem varies according to the heuristics’ combinations. The cost calculation sub-problem has negligible complexity.

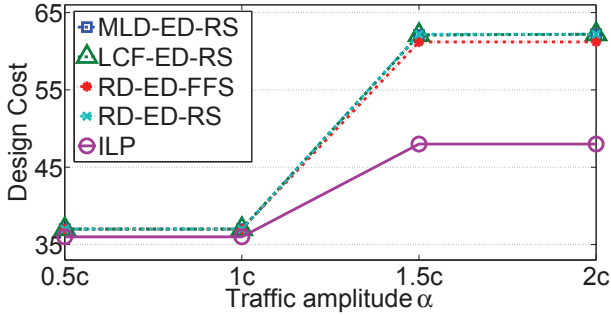


Fig. 3. Design cost in case of distributed traffic load.

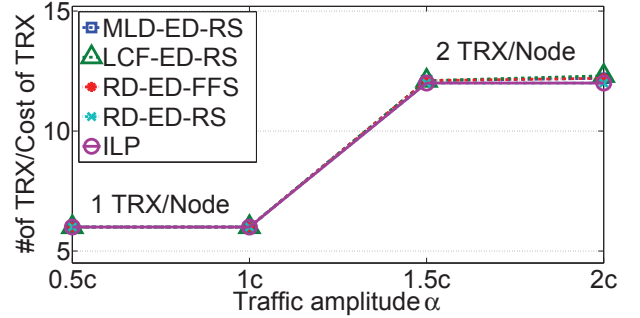


Fig. 4. Transponders cost in case of distributed traffic load.

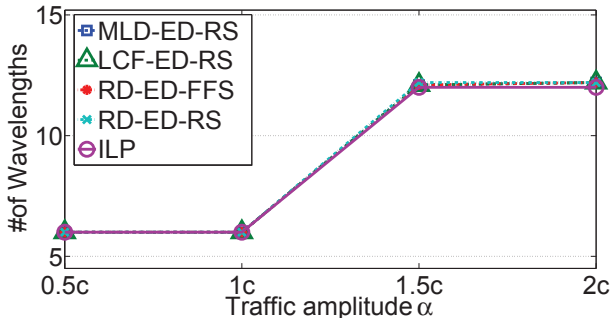


Fig. 5. Number of wavelengths in case of distributed traffic load.

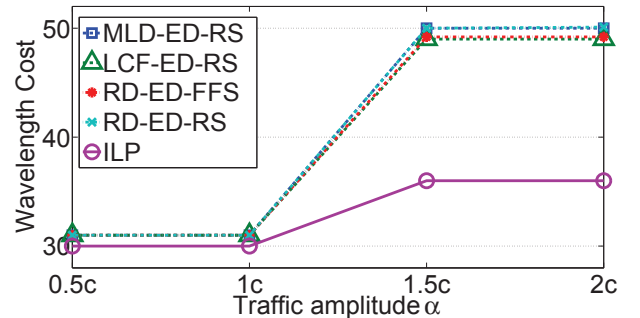


Fig. 6. Wavelength cost in case of distributed traffic load.

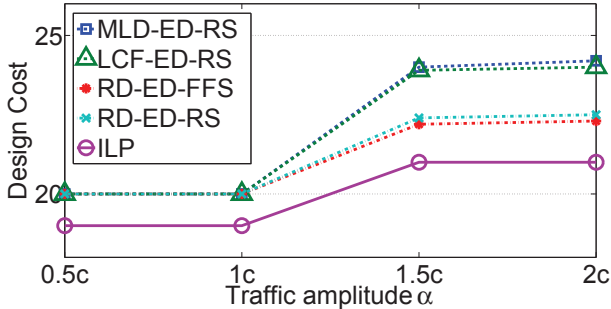


Fig. 7. Design cost in case of centralized traffic load.

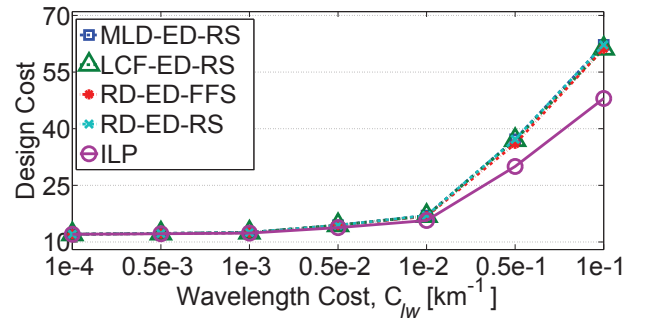


Fig. 8. Network design cost (for distributed input traffic) in a function of the wavelength cost

HMVTD resolves the three sub-problem sequentially, therefore, its global complexity is dominated by the resource allocation algorithm that has the highest complexity. All of the algorithms have a polynomial complexity of a degree between 6 and 8. As in a real network  $K \gg V$ , the algorithms with lowest order of  $K$  (e.g., MLC-ED-FFS and LCF-ED-FFS) are less complex than the others.

In the following, we carry out a detailed study of the performance considering the most pertinent heuristics having the cost within 30% of the optimal solution.

### B. Impact of traffic intensity on the network cost efficiency

In this section we assess the total design cost as a function of traffic intensity with fixed TRX and wavelength cost. We set the TRX cost to  $C_t = 1$  (a.u.). For such  $C_t$ , the wavelength cost is taken at its upper bound, and is fixed to  $C_{lw} = 0.1$  (a.u.)  $\text{km}^{-1}$ . In the following, distributed and centralized traffic are considered.

1) *Distributed Traffic*: The traffic matrix is assumed to be uniform and complete as in the previous section (i.e., there are

30 traffic flows in the network). The total sent/received traffic by a node has the amplitude  $\alpha$ .

The design cost is plotted in Fig. 3. For traffic loads between  $0.5c$  and  $c$ , the design cost is close to the optimal solution. For traffic higher than  $c$ , the design cost is within 27% that of the optimal solution. To explain this difference, we plot the TRX cost and the wavelength cost separately.

Figs. 4 and 5 show that the number of transponders and wavelengths respectively, are very close to the optimal solution. However, Fig. 6 illustrates that the difference in design cost between heuristics and ILP is mainly due to difference in wavelength cost, especially when traffic exceeds the wavelength capacity ( $\alpha > c$ ). The penalty due to the wavelength cost is the result of the simplified wavelength assignment process. Indeed, when a wavelength is completely filled in, HMVTD assigns another wavelength and continues the resource allocation process for the remaining demands without minimizing the total wavelength per fiber length. The chosen wavelength assignment process helps to significantly reduce the heuristic complexity since it avoids performing the

resource allocation process from the beginning each time a new wavelength is needed. However, this is achieved at price of the increased total network cost.

2) *Centralized Traffic*: Centralized traffic is the case where one of the nodes in the network plays the role of a gateway to the backbone network. In our example, the gateway is located at node A, and it is supposed that this node sends and receives back the traffic with the same amplitude from all the nodes in the network (i.e.,  $\alpha$  is the total sent/received traffic amplitude by the gateway). This scenario is close to the current operation of metropolitan networks.

The design cost comparison is presented in Fig. 7. The combinations using the RD policy have almost the same performance, with design cost within 6% that of the optimal solution. The MLD and LCF policies have similar performance and the average design cost is within 15% that of the optimal solution for  $\alpha = 2c$ . Similar to the distributed case, the cost penalty comes from the wavelength leasing cost over the network links.

### C. The impact of the wavelength cost on the network cost

The previous simulations show that the penalty in design cost (w.r.t. the optimal solution) is mainly due to the wavelength cost. In order to get a more complete understanding of the behavior for different policies, we study the impact of the wavelength cost  $C_{lw}$  on the design cost. We consider for this study a distributed traffic profile, with fixed amplitude  $\alpha = 2c$  as at this load the heuristics' cost penalty is the most significant. The results plotted in Fig. 8 show that for small values of wavelength cost, all four policies are very close to the optimal solution. However, for the sufficiently high value of  $C_{lw}$  (greater than 1% of the TRX cost), the design cost penalty of the heuristics increases, until it reaches 27% of the optimal solution.

### D. The impact of virtualization on the network cost

In this section, we assess the performance of the HMVTD for multiple virtualization domains using the RD-ED-FFS heuristic combination, that has the best performance according to the previous study. The distributed scenario with traffic amplitude  $\alpha = 2c$  is used.

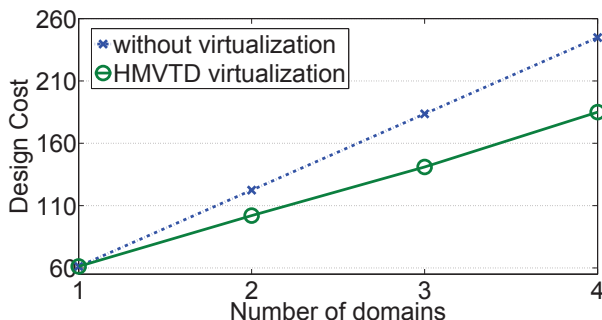


Fig. 9. The impact of virtualization: Design cost in case of distributed traffic load.

Fig. 9 shows that sharing the resources between multiple virtualization domains as defined by HMVTD leads to up to

30% savings in design cost comparing to the network dimensioning without virtualization, when resources are separately allocated for each domain.

## IV. CONCLUSION

In this paper, we have presented the comprehensive algorithm called HMVTD (Heuristic Method for Virtual TWIN Dimensioning), that is a scalable solution for routing, scheduling and virtualization in TWIN optical burst switching network. Results show that the most pertinent heuristics can reach a cost within 27% that of the optimal solution. This penalty is mainly due to the wavelength leasing cost, since the number of used transponders and wavelengths are close to the optimal. We also demonstrated that the impact of the wavelength leasing cost is dominant when the wavelength cost per km is superior to 1% of the transponder cost. In addition, applying HMVTD to network virtualization leads to 30% saving in design cost w.r.t. the case without virtualization.

## ACKNOWLEDGMENT

This work was partly supported by the DGE, in the frame of the CELTIC-Plus project SASER-SaveNet.

## REFERENCES

- [1] I. Widjaja, I. Saniee, R. Giles, and D. Mitra, "Light core and intelligent edge for a flexible, thin-layered, and cost-effective optical transport network," *Communications Magazine, IEEE*, vol. 41, no. 5, pp. S30–S36, 2003.
- [2] R. Indre, J. Pesic, and J. Roberts, "POPI: A passive optical pod interconnect for high performance data centers," in *Optical Network Design and Modeling, International Conference on*, 2014, pp. 84–89.
- [3] K. Ross, N. Bambos, K. Kumaran, I. Saniee, and I. Widjaja, "Scheduling bursts in time-domain wavelength interleaved networks," *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 9, pp. 1441–1451, 2003.
- [4] I. Saniee and I. Widjaja, "Design and performance of randomized schedules for time-domain wavelength interleaved networks," *Bell Labs Technical Journal*, vol. 2, no. 14, pp. 97–112, 2009.
- [5] A. Triki, R. Aparicio-Pardo, P. Gavignet, E. Le Rouzic, B. Arzur, and A. Gravey, "Is it worth adapting sub-wavelength switching control plane to traffic variations?" in *Optical Network Design and Modeling, International Conference on*. IEEE, 2014, pp. 186–191.
- [6] B. Ušćumlić, Y. Pointurier, A. Gravey, P. Gravey, and L. Alahdab, "Scheduling aware dimensioning for time-domain wavelength interleaved network," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*. IEEE, 2014, pp. 484–490.
- [7] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *CoRR*, vol. abs/1406.0440, 2014. [Online]. Available: <http://arxiv.org/abs/1406.0440>
- [8] M. Jinno, H. Takara, K. Yonenaga, and A. Hirano, "Virtualization in optical networks from network level to hardware level [invited]," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 5, no. 10, pp. A46–A56, Oct 2013.
- [9] R. Nejabati, E. Escalona, S. Peng, and D. Simeonidou, "Optical network virtualization," in *Optical Network Design and Modeling, International Conference on*, 2011, pp. 1–5.
- [10] M. I. Kamel, L. B. Le, and A. Girard, "LTE wireless network virtualization: Dynamic slicing via flexible scheduling," in  *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*. IEEE, 2014, pp. 1–5.
- [11] A. Kershbaum and R. Van Slyke, "Computing minimum spanning trees efficiently," in *Proceedings of the ACM annual conference-Volume 1*. ACM, 1972, pp. 518–527.
- [12] B. Ušćumlić, I. Cerutti, A. Gravey, P. Gravey, D. Barth, M. Morvan, and P. Castoldi, "Optimal dimensioning of the WDM unidirectional ecoframe optical packet ring," *PNC*, vol. 22, no. 3, pp. 254–265, July 2011.
- [13] G. N. Frederickson, "A single source shortest path algorithm for a planar distributed network," in *STACS 85*. Springer, 1985, pp. 143–150.