An educational RWA network planning tool for dynamic flows

P. Pavón-Mariño, B. García-Manrubia, R. Aparicio-Pardo, J. García-Haro, G. Moreno-Muñoz

Abstract—**This paper presents the Dyn-MatPlanWDM tool, an educational network planning tool for wavelength-routing WDM networks. The tool considers the on-line or dynamic network operation. Traffic is composed of higher level traffic connections. The planning algorithms dynamically decide on the virtual topology design and the routing of the flows on top of it. The input parameters to the planning problem are the network physical topology, technological constraints like the number of transmitters, receivers, optical converters, the generator module and the planning module to test. The tool is implemented as a MATLAB toolbox. The set of generators and planning modules included can be easily extended. In particular, a set of libraries are provided to ease the composition of new heuristic algorithms. A Graphical User Interface is supplied which aids the user to test and trace the network evolution. The GUI is designed to support an educational use of the tool.**

Index Terms— **Wavelength-routing networks, network planning, educational tool, computer network performance.**

I. Introduction

Wavelength Routing (WR) networks are designed to alleviate the gap between the transmission capacities available in the Wavelength Division Multiplexing (WDM) links, and the inferior switching capacity of the electronic switching nodes. In WR networks, traffic is carried onto transparent lightpaths that may traverse a variable number of nodes. This is performed by reconfigurable optical add/drop multiplexers (R-OADM) or reconfigurable wavelength cross-connects (R-WXC), which allow some wavelengths to be dropped or added in a link, while others are optically switched without electronic conversion [1]. By allowing a transparent optical switching of a fraction of the traffic across the switching nodes, the electronic switching bottleneck is lighten.

Fig. 1 sketches a node of this type. Fig. 1(a) identifies the two types of optically switched lightpaths without (1) or with (2) wavelength conversion. Fig. 1(b) shows four lightpaths which carry traffic that is processed electronically in the G-fabric (grooming fabric) of the node. The 100% of the traffic in lightpath (3) is ingress traffic added by the node. The 100% of the traffic in lightpath (4) is egress traffic dropped in the node. Lightpaths (5) and (6) show a much more common situation. Some portion of the traffic in lightpath (5) is dropped in the node. The rest of the traffic is groomed. That is, it is allocated in other lightpaths like (6) which are initiated in this node, sharing the lightpath with other grooming and/or add traffic.

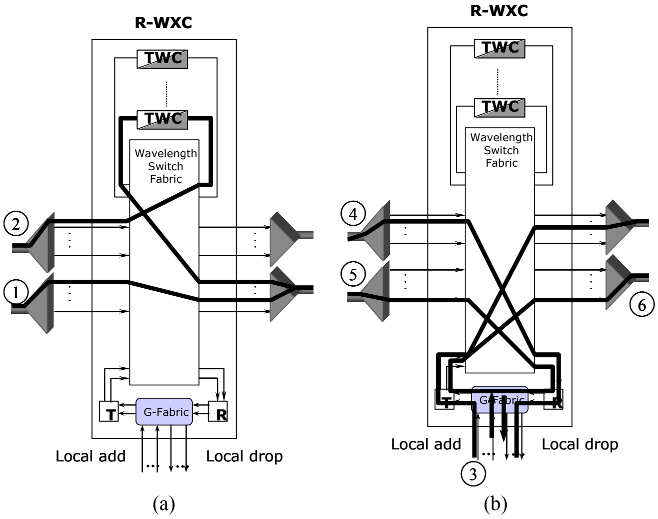


Fig. 1. Functional blocks in the switching node

Fig. 1 helps us to identify the main functional blocks in the switching nodes which impact the overall cost of the network: a) the number of input and output wavelengths, which may be different for each fiber link, b) the number of electro-optic (E/O) transmitters (T) and opto-electronic (O/E) receivers (R) in the node, c) the number of tunable wavelength converters (TWC) in the node and d) the electronic switching capacity required, given by the sum of the ingress, egress and grooming traffic.

The actual equipment required in a network depends on the particular lightpaths established and their route on top of the physical topology. This is called, the virtual topology design. Two types of optimization strategies exist when addressing the network planning process. They are commonly called as static vs. dynamic planning (also off-line vs on-line planning).

Manuscript received November 19, 2007. This research has been supported by project grants TEC 2004-05622-C04-02/TCM (ARPAq), TEC2007-67966-01/TCM (CON-PARTE-1) and it is also developed in the framework of "Programa de Ayudas a Grupos de Excelencia de la Región de Murcia, de la Fundación Séneca, Agencia de Ciencia y Tecnología de la RM (Plan Regional de Ciencia y Tecnología 2007/2010). The authors participate in the COST 291 action, TEC2006-27633-E and TEC2006-27651-E Spanish thematic networks, e-Photon/ONe+ and BONE EU networks of excellence.

Authors are with the Universidad Politécnica de Cartagena (Spain). (corresponding author Pablo Pavón Mariño: +34 968325952, e-mail: Pablo.Pavon@ upct.es).

In static planning, traffic demand is modeled as an *N*x*N* traffic matrix, being *N* the number of nodes in the network. Each coordinate (*i*,*j*) collects the total expected traffic from node *i* to node *j*. In the dynamic planning, the traffic demand is modeled at a finer granularity, as a *temporal* sequence of individual traffic connection requests of higher level flows. The dynamic planning involves taking a decision every time a new flow demand arrives, and possibly every time an existing flow finishes. In the general case, new flows may be allocated in existing lightpaths or involve the establishment of one or more new ones. Also, new flows may imply termination of reallocation of existing lightpaths. Although not common, a flow termination may involve the reallocation or termination of existing flows and/or lightpaths. If compared to static planning, dynamic planning allows a finer evaluation of the network performances, and the observation of realistic performance indicators as the blocking probability of the flow demands.

This paper presents Dyn-MatPlanWDM tool: an educational simulation tool for dynamic planning of WR networks with or without the wavelength continuity constraint. The tool automates the performance evaluation of *dynamic* optimization algorithms. The tool is implemented as a MATLAB toolbox, together with a Graphical User Interface (GUI). It is designed to be easily extended in three manners: new optimization algorithms, new traffic generation modules, and new physical topologies.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 discusses the toolbox structure. Then, the GUI is presented in section 4, next to a set of exemplifying results. Finally, section 5 concludes.

II. RELATED WORK

A. Planning of dynamic flows

On-line optimization algorithms react to higher level traffic connections arrivals and terminations. The overall problem implies deciding the evolution of the virtual topology and the routing of the flows. Heuristic algorithms proposed for this problem are commonly composed of other heuristic algorithms that address planning sub-problems. For instance:

- Lightpath routing algorithms: For a lightpath request between an ingress-egress node pair and for a given physical topology, the algorithm computes the path of physical hops for the lightpath. See for example: [1], [2], [3] and [4].
- Wavelength assignment algorithms (WA): For a lightpath request between an ingress-egress node pair in a given physical topology, and given the route of the lightpath, the algorithm selects the wavelength assignment in each hop with or without the wavelength continuity constraint. See for example: [1], [2], [5] and [6].
- Routing and Wavelength Assignment algorithms (RWA): For a lightpath request between an ingress-egress node pair, in a given physical topology, the algorithm calculates a route for the lightpath including the wavelength assignment in each hop, with or without the wavelength continuity constraint. See for example: [1], [6], [7], [8] and [9].
- Rerouting and Wavelength Assignment algorithms (RRWA): This problem is similar to the RWA one. The only difference lays on the possibility to reroute existing lightpaths in the network, in order to accommodate the new lightpath request. This type of algorithms is commonly designed for WR networks with the wavelength continuity constraint, and search for a solution which minimize the rerouting cost. See for example: [1] and [10].
- Flow routing algorithms. For a flow request between an ingress-egress node pair and for a given virtual topology, it computes the path of virtual hops for the flow. See for example: [11] and [12].

B. Dynamic planning tools

This section briefly overviews some relevant educational planning tools for dynamic flows. The GMPLS Lightwave Agile Switching Simulator (GLASS) [13] is an educational tool built as an evolution of the MERLiN (Modeling Evaluation and Research of Lightwave Network) project, which was discontinued in 2001 [14]. The GLASS tool was developed at the National Institute of Standards and Technology (NIST). It is implemented in Java, and designed on top of the SSF/SSFNet [13] discrete event simulation framework. GLASS uses the Data Modeling Language (DML) [13] to design the topology and derive scripts for the different simulation scenarios and components. The GLASS core is designed as a command-line simulator that reads its simulation setup from a script and writes dumps in binary files that can be processed by customized readers. The Towson State University developed a graphical Topology and Simulation Creator (TSC) to lessen this burden. GLASS is of free use, and its last public version update occurred in March 2006. The Dyn-MatPlanWDM approach offers some advantages when compared to GLASS tool. First, GLASS requires training in a complex class hierarchy of Java classes to allow the definition and testing of new optimization algorithms. Second, Java language does not offer the mathematical functionality that MATLAB provides for a fast implementation of complex algorithms.

The *Optical WDM Network Simulator* (*OWns*) [15] tool, was developed at the Washington University, as a tool built on top of the well-known network simulator *ns*. The *OWns* facilitates the study of RWA dynamic algorithms in WDM networks. Unfortunately, the tool upgrading stopped in 2001. When compared to Dyn-MatPlanWDM, the benefits of our proposal are again based on the more powerful framework that MATLAB provides, which further helps to simplify the algorithm implementation and clarify to the student the planning concepts behind.

Of course, a wide range of commercial utilities also exists, for the dimensioning of WDM networks under dynamic traffic [16-18]. Nevertheless, these tools are not designed for educational purposes: their underlying algorithmic details are not publicly available, and can not be extended by the researchers or students. Therefore, they are out of the scope of this survey.

III. DESCRIPTION OF THE TOOLBOX

The Dyn-MatPlanWDM tool has been implemented as a MATLAB [15] toolbox. All the functions in the toolbox are documented, and open for free usage. Fig. 2 helps us to describe the structure of the toolbox.

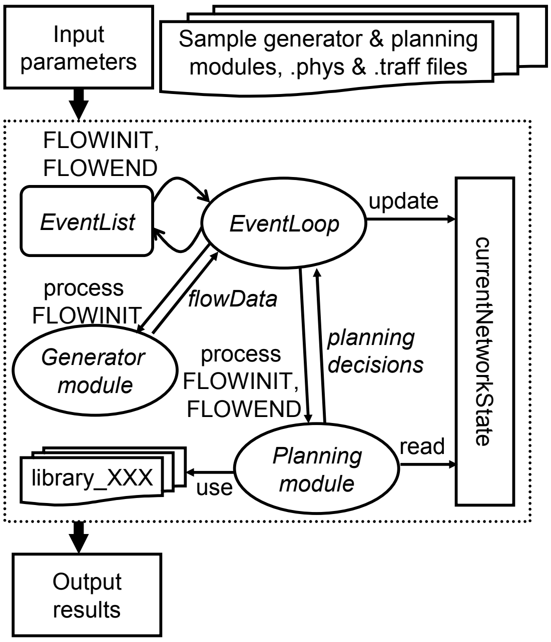


Fig. 2. Dyn-MatPlanWDM toolbox diagram.

A. Input parameters

The input parameters for the simulation are specified in three files: (1) a text file describing the physical topology, establishing the information per node (number of E/O transmitters, O/E receivers, tunable wavelength converters -TWC-), the number of wavelengths per link, and the lightpath capacity (in Gbps). (2) A MATLAB file which implements the generator module, and (3) a MATLAB file which implements the planning module. User defined input parameters can be passed to the two latter modules for initialization.

B. Event loop

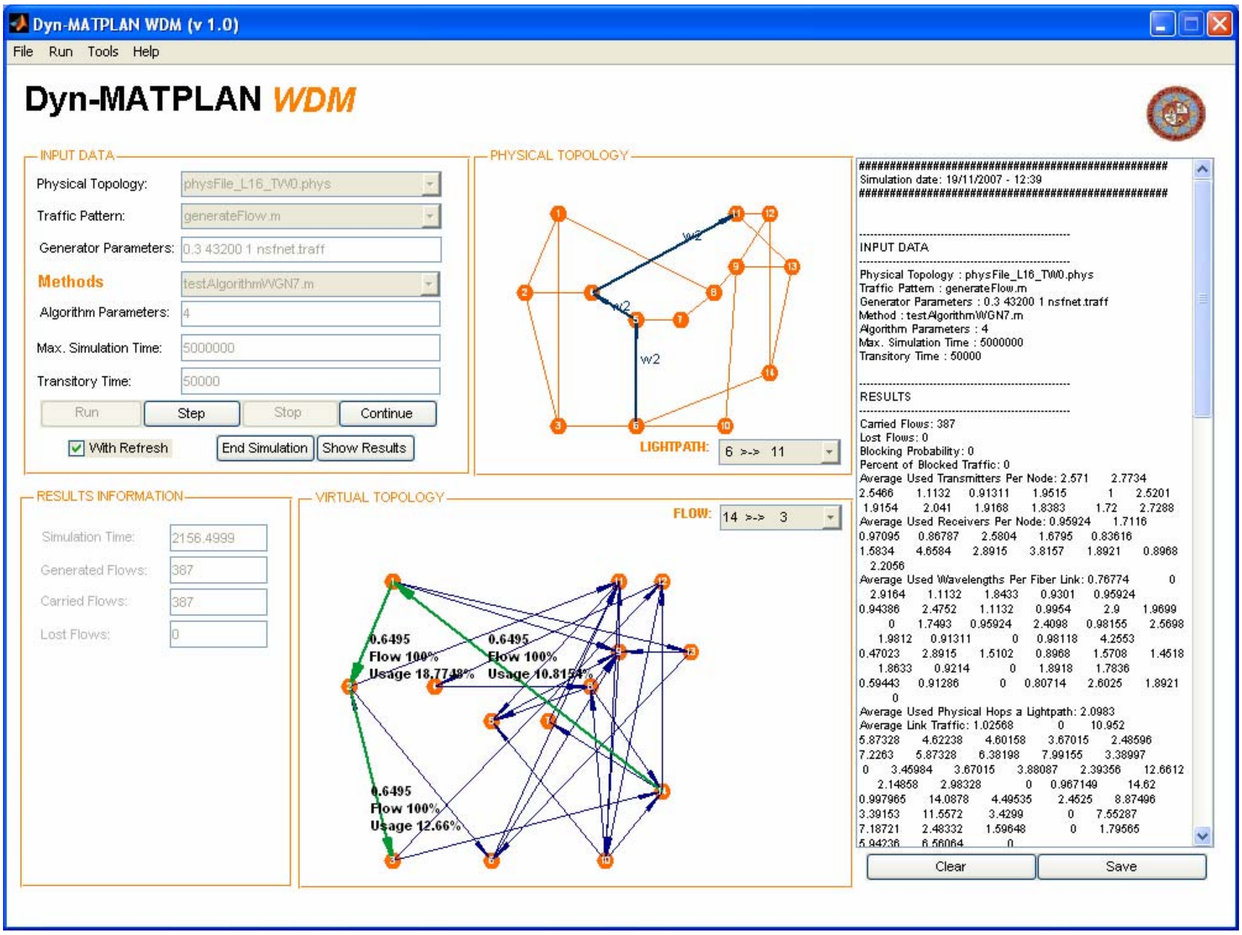
The *EventLoop* module is the engine of the simulator. It controls an event list, being the only module capable of adding and removing events from the list. The list contains just two types of events: (i) new flow arrival (FLOWINIT), (ii) and existing flow termination (FLOWEND). These are processed as follows:

- FLOWINIT: When processing this event, the *Event Loop* first invokes the *Generator* module. The module should return the arriving flow *flowData* information (flow ingress and egress node, flow average rate in Gbps, flow duration in seconds, flow priority) and the time when the next flow should arrive (*timeToNextFlow*). The *timeToNextFlow* information is used by the *EventLoop* to add a new FLOWINIT event in the event list. This schedules the next flow arrival. Then, the *EventLoop* calls the *Planning* module, passing the *flowData* information as a parameter, along with the *currentNetState* structure. The planning algorithm decides whether the new flow can be carried or not, and the *ordered sequence* of planning decisions to take in order to accommodate the flow. Each action can be a lightpath setup, termination or reroute, and a flow setup, termination or reroute. Note that this paradigm is fully flexible and allows the testing of *Planning* modules were older flows can be reallocated or removed according to its priority, releasing resources to carry new flows, or modifying the virtual topology as desired. The information returned by the planning algorithm is used by the *EventModule* to update the *currentNetworkState* structure and the simulation statistics accordingly. If the planning algorithm decided to carry the arriving flow, a FLOWEND event is added to the event list to schedule that flow termination.
- FLOWEND: When processing this event, the *Event Loop* invokes the *Planning* module. This module is responsible for deciding the planning decisions to be taken, returned as a set of actions similarly as with the FLOWINIT events.

This *currentNetState* structure is a set of variables which describes the current network state. The *EventTool* updates this structure in every event cycle according to the actions planned by the *Planning* module. The *Planning* modules can access the *currentNetState* structure to obtain the desired information regarding the existing lightpaths and their routes, the carried flows and their routes, and the available resources per link (wavelengths) and node (transmitters, receivers and TWCs).

C. Output results

The tool automates the calculation of a large set of statistics, which can be stored in a text file for further processing. The tool returns the minimum, maximum and



Physical Topology: physFile_L16_TW0.phys

Traffic Pattern: generateFlow.m

Generator Parameters: 0.3 43200 1 nsfnet.traff

Methods testAlgorithmWGN7.m

Algorithm Parameters: 4

Max. Simulation Time: 5000000

Transitory Time: 50000

☑ With Refresh

LIGHTPATH: 6 >-> 11

FLOW: 14 >-> 3

Simulation Time: 2156.4999

Generated Flows: 387

Carried Flows: 387

Lost Flows: 0

```
#####################################################
Simulation date: 19/11/2007 - 12:39
#####################################################


-------------------------------------------------
INPUT DATA
-------------------------------------------------
Physical Topology : physFile_L16_TW0.phys
Traffic Pattern : generateFlow.m
Generator Parameters : 0.3 43200 1 nsfnet.traff
Method : testAlgorithmWGN7.m
Algorithm Parameters : 4
Max. Simulation Time : 5000000
Transitory Time : 50000

-------------------------------------------------
RESULTS
-------------------------------------------------
Carried Flows: 387
Lost Flows: 0
Blocking Probability: 0
Percent of Blocked Traffic: 0
Average Used Transmitters Per Node: 2.571     2.7734
2.5466     1.1132    0.91311     1.9515          1     2.5201
 1.9154      2.041     1.9168     1.8383       1.72     2.7288
Average Used Receivers Per Node: 0.95924     1.7116
0.97095    0.86787     2.5804     1.6795    0.83616
1.5834    4.6584     2.8915     3.8157     1.8921     0.8968
  2.2056
Average Used Wavelengths Per Fiber Link: 0.76774         0
  2.9164     1.1132     1.8433     0.9301    0.95924
0.94386     2.4752     1.1132     0.9954        2.9     1.9699
      0     1.7493    0.95924     2.4098    0.98155     2.5698
   1.9812    0.91311          0    0.98118     4.2553
0.47023     2.8915     1.5102     0.8968     1.5708     1.4518
   1.8633     0.9214          0     1.8918     1.7836
0.59443    0.91286          0    0.80714     2.6025     1.8921
      0
Average Used Physical Hops a Lightpath: 2.0983
Average Link Traffic: 1.02568          0     10.952
5.87328     4.62238     4.60158     3.67015     2.48596
7.2263     5.87328     6.38198     7.99155     3.38997
 0     3.45984     3.67015     3.88087     2.39356    12.6612
   2.14858     2.98328          0    0.967149     14.62
0.997965     14.0878     4.49535     2.4525     8.87496
3.39153     11.5572     3.4299          0     7.55287
7.18721     2.48332     1.59648          0     1.79565
5.94236     6.56064          0
```

Fig. 3. Dyn-MatPlanWDM Graphical User Interface main window.

temporal average value of the following parameters:

- Associated to the cost of the network: number of used O/E, E/O, and TWCs per node, number of used wavelengths per fiber, amount of traffic electronically processed in each node (Gbps).
- Associated to the performance perceived by the carried traffic: average number of physical hops of the lightpaths on top of the links, average number of virtual hops of the flows on top of the lightpaths, single virtual hop traffic, average message propagation, network congestion (defined as the traffic utilization of the lightpath with the highest traffic utilization).
- Associated to the blocking performance: blocking probability, percentage of blocked traffic.

D. *Extension of the tool and help libraries*

The tool can be easily extended in different ways. Any physical topology can be tested by creating the appropriate *.phys* file, following the instructions in the tool, and saving it in the designated directory. In addition, new generator modules and new planning algorithms can be implemented as MATLAB functions with a given signature. Integration of the modules simply requires saving them in the designated directory.

Dyn-MatPlanWDM tool includes a set of help libraries composed of MATLAB functions, suitable for the construction of planning modules:

- *library_Graph*: Includes general algorithms in graphs. This version of the tool includes algorithms for solving the shortest path problem, the capacitated shortest path problem and the *k*-shortest path problem following the path removal algorithm in [19].
- *library_WA*: Includes heuristics for the WA subproblem. Two algorithms are implemented nowadays. One of them is the first-fit [1] algorithm for non-wavelength convertible networks. The other one is a general graph conversion technique similar to the LEast Converter First (LEC) and Least Conversion Cost First (LCC), both ones used in [6], and the graph conversion technique used in [8].
- *library_RWA*: Includes heuristics for the RWA subproblem. The heuristics implemented at this moment are the Joint Wavelength-Route Selection (JWRS) [1] and the Wavelength- and Converter-

Aware Algorithm (WCA) [6].

- *library_RRWA*: Includes heuristics for the RRWA subproblem. The heuristics currently implemented are the Auxiliary Graph (AG) algorithm [1] and [10].

IV. Graphical User Interface

The tool provides two ways of launching simulations: a batch simulation module and a Graphical User Interface (GUI). The batch simulation module is a script which receives a range of input parameters to be tested, sequentially executes the simulations, and organizes the output result files in a given directory.

The GUI is specifically designed for educational purposes. Its objective is to give the students the opportunity to test their own planning algorithms. It allows tracing the execution of the algorithm event by event, observing graphically the evolution of the virtual topology, and the routing of the flows on top of it.

Fig. 3 displays the main GUI window workspace, organized under four higher level menus: *File*, *Run*, *Tools* and *Help*. *File* menu aids in the organization of the input and output files. The *Tools* menu provides a set of tools for generating physical topologies, and functions to normalize traffic matrixes [20]. The *Help* menu gives access to the tool documentation. The *Run* menu controls the simulation execution through the workspace shown in Fig. 3. Left-upper corner of the window is devoted to receiving the input parameters from the user: the *.phys* file defining the physical topology, the generator and the planning modules, along with their user defined parameters, the completion time and transitory time of the simulation (in virtual seconds). The physical topology under test is plotted next to the input parameters panel. The user can start the simulation clicking the *Run* button. If the "*with refresh*" checkbox is activated, some information is updated during the simulation: the simulation time, the number of generated, carried and lost flows. The simulation can be stopped with the *Stop* button or the *End Simulation* button. The former, stops the simulations but does not end it. At this moment, the *Step* button is enabled, giving the possibility to trace the execution consuming one event every time the *Step* button is pressed. Then, the state of the network can be observed in a graphical way in the physical topology panel (center-up), and virtual topology panel (center-down). The user may employ the option menu in the physical topology panel, to access the RWA information of a particular lightpath. In the Fig. 3, a lightpath from node 6 to node 11 is chosen, and its hop sequence {6,5,4,11}, and wavelength sequence $\{w_1, w_1, w_1\}$ are plotted.

The virtual topology panel draws one arc for each existing lightpath. Again, the user may make use of the option menu in that panel to access the routing information of a particular carried flow. In the figure, a flow from 14 to node 3 is chosen, and its traversing lightpath sequence {(14,1);(1,2);(2,3)} is highlighted. The flow rate in Gbps, and the fraction of traffic in the lightpath corresponding to the particular flow are also shown (i.e. 100% if it is the unique flow in that lightpath).

The rightmost panel is a text area where the simulation input parameters and results are written. The user can save this file appropriately for a further processing.

A. Sample results

This section collects and presents some results to illustrate the utilization of the tool. The physical topology tested is the 14-node NSFNET network, with 8 wavelengths per link, no wavelength conversion, and an infinite number of E/O transmitters and O/E receivers per node, all of them of 10 Gbps. The generator module employed, models the arrivals of traffic flows of an exponentially distributed duration (average duration of 12 hours) and an exponentially distributed rate (average rate of 300 Mbps). The time between arrivals of two flows from the same ingress node to the same egress node, is also exponentially distributed. The average interarrival time of flows between node i and j, can be different for each (i,j) pair. It is selected so that the average traffic injected matches the (i,j) coordinate of a given traffic matrix. Seven traffic matrixes have been tested: $T_{3.2}$, $T_{4.0}$, $T_{4.8}$, $T_{5.6}$, $T_{6.4}$, $T_{7.2}$, $T_{8.0}$. All of them are calculated after the traffic matrix proposed for the NSFNET network in [21] (which we denote as T). Traffic matrix T_X is obtained by a normalization process which multiplies the source traffic matrix T, by the factor which makes that the maximum traffic generated from any node i to any node j equals X Gbps.

The planning algorithm used for our test is based on the k-shortest path search (it uses an implementation of this algorithm in its *path removing* version [19] included in the *library_Graph* directory of the toolbox). The planning algorithm operates as follows:

- For a flow arrival: first, it tries to accommodate the flow in the existing virtual topology minimizing the number of virtual hops. It uses the Capacitated Shortest Path algorithm included in the *library_Graph* for that purpose. If it fails, it tries to solve the problem by establishing one lightpath. First, it tries a direct lightpath from ingress to egress node. If it fails, it calculates the set S_{in} of nodes that are connected to the ingress node by a lightpath with enough capacity. Then, for each node in S_{in}, it tries to establish a lightpath from that node to the egress node. If this fails, it computes the set S_{eg} of nodes that are connected to the egress node by a lightpath of enough capacity. For each node in S_{eg} it tries a lightpath from the ingress node to that node. If this fails, the flow is blocked. Each search of a lighpath from an ingress to an egress node is implemented by first obtaining the k-shortest paths in number of physical hops. For each possible route, a free wavelength is selected following the first-fit scheme implemented in the *library_WA*.

- For a flow termination: the planning module removes the flow and checks if, as a consequence, any of the traversing lightpaths becomes empty. The empty lightpaths are then also terminated.

The blocking probability results obtained are plotted in Fig. 4, for the values of *k*={1,2,3,4}. Simulation virtual time was fixed to 1 month. Results measure the expected improvement of the blocking performance in the network for higher values of parameter *k*. Also, as expected, the *k*-based improvement of the blocking probabilities shrinks at higher loads of the network.

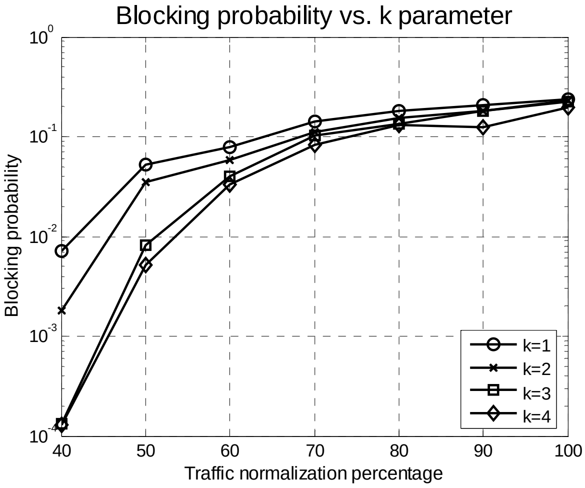


Fig. 4. Blocking probability performance of the tested scenario, for the values *k*={1,2,3,4}, and normalized traffic matrixes $T_{3.2}$, $T_{4.0}$, $T_{4.8}$, $T_{5.6}$, $T_{6.4}$, $T_{7.2}$, $T_{8.0}$.

V. Conclusions and Further Work

This paper presents a MATLAB-based educational planning tool for WR networks. The tool is easily extensible and fully documented. It includes a set of libraries to ease the composition of new heuristic algorithms, which can be tested or traced with the GUI. The tool is well-suited for being used at research, and at postgraduate level courses.

At this moment, the Dyn-MatPlanWDM tool is being integrated in the already existing MatPlanWDM tool [22] (publicly available at MatlabCentral web site, http://www.matlabcentral.com), developed by the same group, which focuses on off-line or static planning.

References

[1] C.S.R. Murthy and M. Gurusamy, *WDM Optical Networks (Concepts, Design and Algorithms)*. Upper Sadle River, NJ: Prentice Hall PTR, 2002, ch 2.

[2] H. Zang, J.P. Jue and B. Mukherjeee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, vol. 1, no. 1, pp 47-60, Jan. 2000.

[3] X. Chu, B. Li, Z. Zhang, "A Dynamic RWA Algorithm in a Wavelength-Routed All-Optical Network with Wavelength Converters," in *22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, Mar. 2003, vol. 3, pp.1795-1804

[4] B. Li, X. Chu, K. Sohraby, "Routing and wavelength assignment versus wavelength converters placement in all-optical networks," *IEEE Communications Magazine*, vol. 41, no. 8, pp. S22-S28, Aug. 2003

[5] X. Zhang, C. Qiao, "Wavelength Assignment for Dynamic Traffic in Multi-Fiber WDM Networks," in *7th International Conference on Computer Communications and Networks (ICCCN '98)*, 1998, pp. 479-485

[6] Y. Zhang, K. Taira, H. Takagi, S.K. Das, "An efficient Heuristic for Routing and Wavelength Assignment in Optical WDM Networks," *IEEE International Conference on Communications*, vol. 5, pp. 2734- 2739 , 2002

[7] T. Fabry-Asztalos, N.M. Bhide, K.M Sivalingam, "Routing Mechanisms Employing Adaptative Weight Functions for Shortest Path Routing in Optical WDM Networks," *Photonic Network Communications*, vol. 3, no. 3, pp. 227-236, Jul. 2001

[8] K.C. Lee and V.O.K. Li. "A wavelength-convertible optical network," *Journal of Lightwave Technology*, vol. 11, no. 5/6, pp. 962-970, May/Jun. 1993

[9] Z. Zhang, A.S. Acampora, "A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength re-use," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 281-288, Jun. 1995.

[10] C.S.R. Murthy and M. Gurusamy, "A Time Optimal Wavelength Rerouting Algorithm for Dynamic Traffic in WDM Networks," *Journal of Lightwave Technology* , vol. 17, no. 3, pp 406-417, Mar. 1999

[11] R. Ramaswami, K.N. Sivarajan, "Design of Logical Topologies for Wavelength-routed Optical Networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp 840–851, Jun 1996

[12] V.R. Konda, T.Y. Chow, "Algorithm for Traffic Grooming in Optical Networks to Minimize the Number of Transceivers," *IEEE Workshop on High Performance Switching and Routing*, pp. 218-221, 2001

[13] http://www-x.antd.nist.gov/glass/ (last access: 19th November, 2007)

[14] F. Mouveaux, F. Lapeyrere, N. Golmie. (2000, February). *MERLIN, User's Manual and Programmer's Guide*. Vs 1.0. National Institute of Standards and Technology.
Available: http://w3.antd.nist.gov/Hsntg/prd_merlin.html.

[15] B. Wen et al., "Optical Wavelength Division Multiplexing (WDM) Network Simulator (OWns): Architecture and Performance Studies," *SPIE Optical Networks Magazine Special Issue on "Simulation, CAD, and Measurement of Optical Networks"*, Mar. 2001.

[16] http://www.opnet.com (last access: 19th November, 2007)

[17] http://web1.rsoftdesign.com/products/network_modeling/MetroWAND/ (last access: 19th November, 2007)

[18] http://www.optiwave.co.kr/product/optisystem.htm (last access: 19th November, 2007)

[19] E.Q. Martins, "An algorithm to ranking paths that may contains cycles," *European Journal of Operational Research*, vol. 18, pp. 123-130, 1984

[20] R.S. Cahn, *Wide Area Network design. Concepts and tools for optimization.* San Francisco, CA: Morgan Kaufmann Publishers Inc.,1998

[21] R. Ramaswami, K.N. Sivarajan, "Design of Logical Topologies for Wavelength-routed Optical Networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 840–851, Jun. 1996

[22] P. Pavon-Mariño, R. Aparicio-Pardo, G. Moreno-Muñoz, J. Garcia-Haro, J. Veiga-Gontan, "MatPlanWDM: An educational tool for network planning in wavelength-routing networks", Lecture Notes in Computer Science (Springer-Verlag GmbH), vol. 4534, pp. 58-67, May 2007.