

Educational and research tools for network optimization

Jose-Luis Izquierdo-Zaragoza, Pablo Pavon-Marino

*Universidad Politécnica de Cartagena, Cuartel de Antiguones, Plaza del Hospital 1, 30202 Cartagena, Spain
Tel: (34) 968325952, Fax: (34) 968325973, e-mail: {jose.l.izquierdo, pablo.pavon}@upct.es*

ABSTRACT

This paper presents Net2Plan tool and Java Optimization Modeler (JOM) library and discusses their application as research tools and educational resources in the field of network optimization. Net2Plan is designed to assist users in the definition and comparative evaluation of their original network planning algorithms, as well as in the simulation and testing of connection admission control algorithms, and network protection/restoration schemes. In its turn, JOM is a library which helps to model optimization problems in the Java language and solve them interfacing with integrated linear and non-linear solvers. JOM promotes a fast problem prototyping following a vectorial MATLAB-like syntax. The combined use of Net2Plan and JOM gives users from industry and academia a complete environment to simulate, analyze, dimension, optimize and evaluate the performance of their network designs. Both tools can be publicly downloaded from their respective websites.

Keywords: Network planning and optimization, Modeling framework, Discrete-event simulation, Java.

1. INTRODUCTION

Network optimization is a critical process in the deployment and operation of communication networks. Network engineers must balance user performance with network cost and capabilities. For instance, operators want to keep their resources at high utilization and low cost. However, this may result in unacceptable delays or dropped packets. Hence optimization tools are required to produce such designs.

The planning and optimization tools in the field of communication networks span a broad range of platforms, systems, languages, functionalities and applications. Some of the tools are oriented to the industry, whereas other ones are developed to the academia for educational and research purposes. However, existing tools lack of flexibility for testing different planning algorithms (i.e. users are only allowed to use built-in algorithms) or are not designed for educational purposes (i.e. their underlying algorithmic details are not publicly available). This is very important, since network optimization is more than drag and drop icons and connecting them to build a network.

In this context, our group has developed an open-source network optimization and planning tool called Net2Plan [1]. Net2Plan is a basic network optimization tool shipped under LGPL license, not constrained to any specific network technology (and suitable to any of them), which allows users to test their own algorithms, or use the built-in ones. Network designs produced can be analyzed in the graphical user interface. Automatic report generation tools (with built-in or user-defined reports) can give insight into the network design merits. Two post-analysis simulation tools are also integrated in Net2Plan. (i) The resilience simulator permits evaluating the availability performance of protection and restoration algorithms in the network. (ii) The Connection and Admission Control (CAC) simulator is targeted to analyze the performance of on-line provisioning schemes that allocate resources to incoming connections (e.g. virtual circuits requests, lightpath requests, multimedia sessions...).

In some occasions, network design algorithms require solving one or more formulations (i.e. integer linear programs (ILPs), convex programs...), and interfacing with specialized solver software to obtain the solution to them. The JOM (Java Optimization Modeler) tool, also developed in our group and shipped under LGPL license, is targeted to that use. JOM library allows modeling optimization problems in a high-level MATLAB-like syntax, relying on interfaces to a number of free (i.e. GLPK, IPOPT) and commercial solvers (i.e. CPLEX) to obtain numerical solutions. This paper is focused on the description of the main features of the powerful optimization environment formed by the combination of Net2Plan and JOM.

The rest of the paper is organized as follows. Section 2 provides a brief background in our methodology to model network optimization problems. Section 3 describes the architecture and main features of Net2Plan. Section 4 describes the JOM library. Section 5 demonstrates how a simple network optimization problem can be modeled and solved in a straightforward manner using Net2Plan and JOM. Finally, Section 6 concludes.

2. ADDRESSING NETWORK OPTIMIZATION PROBLEMS

In practical network designs different variables can be involved: node locations, links in the network and their capacities, traffic routing, and so on. Usually, network design problems receive some of this information as input parameters (partial network designs, e.g. network topology and traffic demands) and try to optimize the rest (e.g. capacities in the links and traffic routing) according to a performance merit of interest. Optimization algorithms (or just “algorithms”) are the methods that compute a numerical solution to a given problem instance.

The idea behind Net2Plan is that users can progressively design their networks (see Fig. 1). That is, users can chain successive algorithms, each one completing a part of the network design. For instance, you can start with a

network where only the nodes are defined. Then, an algorithm is used to define the links in the network according to some figure of performance. Afterwards, another algorithm can be run to jointly decide on the capacity in the links and routing of the flows, for a given traffic matrix. As a result, Net2Plan can be a powerful tool for communications network planning courses, since students can see step by step how their designs grow. To achieve such functionality, we define a data storage called network plan which contains all the current state of the network design, so every algorithm receives a network plan (or list of, if multilayer) and returns a modified network plan (or list of).

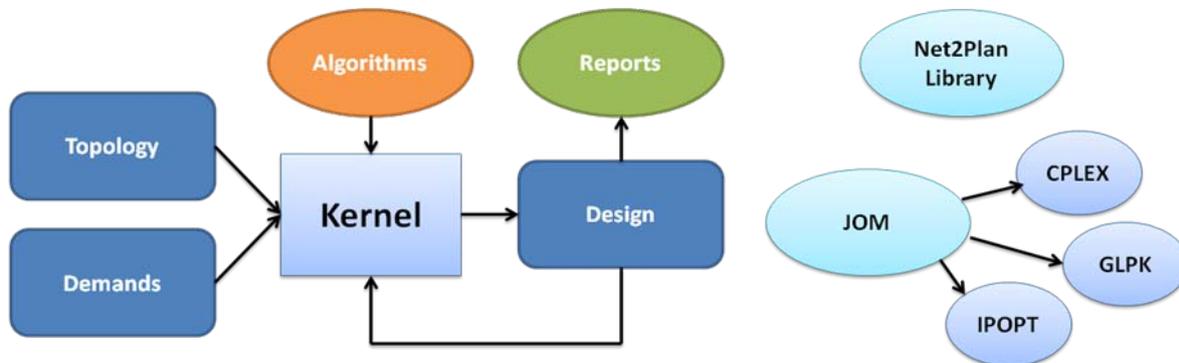


Figure 1. Design flow in Net2Plan.

Net2Plan assists the task of creating and evaluating network design algorithms by providing built-in example algorithms and a set of libraries (e.g. k -loopless shortest paths, minimum spanning tree...). An exhaustive list of built-in algorithms and the Library API Javadoc can be found in [1].

3. NET2PLAN: OPEN-SOURCE NETWORK PLANNER

Net2Plan is designed as an open-source Java application. Net2Plan has its origins in September 2011, as a resource for communications network planning courses at our University. Initial versions were developed in MATLAB (now discontinued), but from February 2013 we switched to Java due to its inherent multi-platform support. Also, no privative software is required at all to run Net2Plan.

The tool is divided into two main parts: user interface (both command-line and graphical user interfaces are provided) and kernel (see Fig. 1). As stated in section 2, the main idea is that all algorithms, independently of their specific target, receive as input parameter the current network design, and return an updated version of this network design. Then, kernel and user interface, are devoted to just process the designs returned by the algorithms: check its validity (e.g. the topology has a correct format, the links are not oversubscribed...), graphically display the design, or compute some reports and performance metrics.

To date, Net2Plan consists of five tools:

- *Network design.* Assists users in the process of network design and planning, by giving access to the execution of built-in or user-defined network design algorithms. The GUI organizes network information, statistics and performance estimations computed from the network design, related to nodes, links, routes, etc. Also, it is possible to generate automatic built-in or user-made reports that estimate more in-depth several network aspects (i.e. network robustness respect to multiple failures, network delay or blocking statistics...).
- *Traffic matrix generation.* Assists users in the process of generating user-defined traffic matrices following or not several models found in the literature [3].
- *Resilience simulation.* Simulates the network operation, where failures in links and nodes randomly appear according to the user-defined availability information of the network nodes/links. Allows testing built-in or user-made provisioning algorithms that react to failure/repairation events from an initial network design. Network and service availability results are collected and reported to the user.
- *Connection-admission-control simulation.* Simulates the network operation, where traffic demands are the source of connection requests. This permits testing the performances of built-in or user-defined CAC (Connection-Admission-Control) algorithms that allocate resources (i.e. the route) to the connections. Network performances i.e. connection blocking are collected and reported to the user.
- *Reporting.* Net2Plan permits the generation of built-in or user-defined reports, from any network design. The report generation tool is integrated within network design, resilience simulation, connection-admission-control simulation tools, so that it is possible to create reports related to any of these network aspects.

4. JOM: JAVA OPTIMIZATION MODELER

Often, some network design problems are solved by modeling them as optimization problems (i.e. integer linear problems, linear problems, convex problems...), and then calling an optimization solver to obtain its numerical solution. In this context, optimization modeling tools are targeted to ease the definition of the problem decision variables, constraints and objective function, and become an interface with the (usually complex) solver libraries.

AMPL and GAMS are examples of commercial modeling tools. Once the decision of migrating Net2Plan to Java was taken, we found that no open-source and free modeling tool existed for Java, and decided to create JOM. JOM is an open-source Java library which can interface with a number of solvers using vectorial MATLAB-like syntax, which i.e. permits the addition of sets of constraints in one line of code. Current JOM version can interface with GPLK (free) and CPLEX (commercial) solvers for mixed integer linear problems, and IPOPT (free) for non-linear differentiable problems. JOM directly interfaces with compiled solver libraries (.DLLs in Windows and .SOs in Linux), via Java Native Access (JNA). JOM is independent from Net2Plan and can be used for any type of optimization problem. However, Net2Plan uses JOM in all the network design algorithms based on solving formulations. A large number of examples using JOM are included in the Net2Plan distribution [1].

In next section, we put together Net2Plan and JOM to solve a network optimization problem.

5. CASE STUDY: SOLVING A MULTI-COMMODITY FLOW ALLOCATION PROBLEM

In this section, we use a classical multi-commodity flow allocation problem [4] to demonstrate how users can model and solve a network optimization problem using Net2Plan and JOM.

Flow allocation problems decide the routing of a set of traffic demands over the physical topology. The objective of the optimization problem is to minimize the maximum link utilization among all links while routes all traffic demands. The entire formulation is shown here:

$$\begin{aligned} \text{Find:} \quad & \rho, x_p && p \in P \\ \text{Maximize:} \quad & \rho \\ \text{Subject to:} \quad & \sum_{d \in D} \sum_{p \in P_d \cap P_e} x_p \leq \rho \cdot u_e && \forall e \in E \end{aligned} \quad (1)$$

$$\sum_{p \in P_d} x_p = h_d \quad \forall d \in D \quad (2)$$

$$0 \leq \rho \leq 1 \quad (3)$$

$$x_p \geq 0 \quad \forall p \in P \quad (4)$$

where $G(N, E)$ is a multi-digraph which represents the physical topology, u_e is the link capacity of link e , D is the traffic demand set, h_d is the average offered traffic volume by demand d , P is the set of candidate paths, P_d is the set of candidate paths allowed to accommodate traffic from demand d ($P_d \subseteq P$), P_e is the set of candidate path traversing link e ($P_e \subseteq P$), ρ is the maximum link utilization, and x_p is the traffic volume carried by each path p .

In Listing 1, a Java code snippet to solve the optimization problem in Net2Plan using JOM and GLPK is shown (see [1] for more information on algorithm implementation). Authors remark the simplicity of the code needed to execute the algorithm and find a solution, compared to alternative solutions using modeling languages such as AMPL, or learn the specific syntax from every solver. Moreover, JOM allows access to the whole Java library and user classes. Finally, we note that the Net2Plan class *CandidatePathList* handles the creation and management of the list of candidate paths for the path-flow formulation.

```
// Compute candidate paths using a K-loopless shortest path algorithm
CandidatePathList cpl = new CandidatePathList(netPlan, "K", "3");
int P = cpl.getNumberOfPaths();

// Create an optimization problem and define input parameters, decision variables,
// constraints and objective function
OptimizationProblem op = new OptimizationProblem();
op.setInputParameter("u_e", netPlan.getLinkCapacityInErlangsVector(), "column");
op.setInputParameter("h_d", netPlan.getDemandOfferedTrafficInErlangsVector(), "column");
op.setInputParameter("delta_dp", cpl.computeDemand2PathAssignmentMatrix());
op.setInputParameter("delta_ep", cpl.computeLink2PathAssignmentMatrix());
op.addDecisionVariable("x_p", false, new int[] {P, 1}, 0, Double.MAX_VALUE);
op.addDecisionVariable("rho", false, new int[] {1, 1}, 0, 1);
op.addConstraint("delta_ep * x_p <= rho * u_e", "capacityConstraints");
op.addConstraint("delta_dp * x_p == h_d", "noLossesConstraints");
op.setObjectiveFunction("minimize", "rho");
```

```

// Solve the problem using a LP solver
op.solve("glpk");
if (!op.solutionIsOptimal()) throw new RuntimeException("Bad");

// Get the solution and include it to the current network design
netPlan.addRoutes(cpl, op.getPrimalSolution("x_p").toDoubleArray(), false);

```

Listing 1. Code snippet to solve the flow allocation problem.

Fig. 2 depicts the network design tool used to solve the flow allocation problem. Users can explore results in a graphical manner, e.g. plotting the route(s) followed by a demand. Results from the above algorithm using the NSFNET reference network (included within Net2Plan) set the traffic routes so that a maximum link utilization of 52.2% is achieved. As an example of data obtained from reports, the average end-to-end delay is 13 ms, and the availability under single and double link-failure using a MPLS Fast Reroute restoration algorithm is equal to 99.99%, for a per-link availability of 99.86% and an independent link failure model.

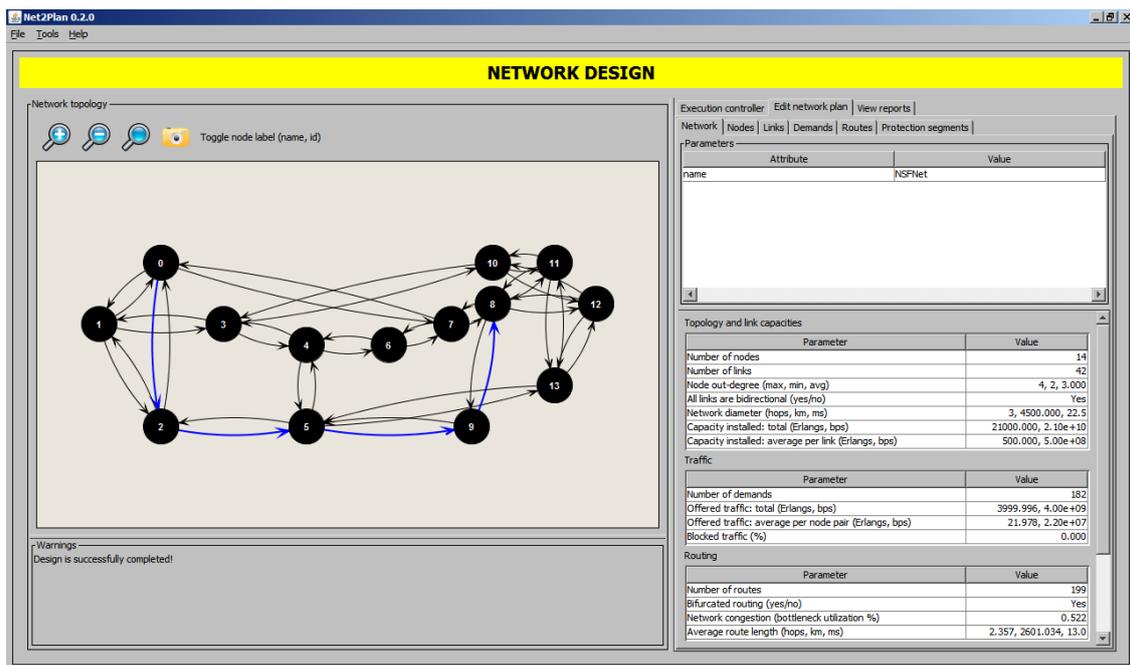


Figure 2. Examining results from a complete network design.

6. CONCLUSIONS

In this paper we present an overview of two open-source tools developed in our University. On the one hand, Net2Plan is a network planning tool which allows users to design, optimize and simulate networks not attached to any specific technology (and thus applicable to any of them), implementing their own algorithms or using the built-in ones. On the other hand, JOM is an optimization modeling library which interfaces with a number of solvers. Combining Net2Plan with JOM, users from research and academia acquire a complete environment for modeling and optimization of communication networks.

ACKNOWLEDGEMENTS

This research was partially supported by the Spanish project grant TEC2010-21405-C02-02/TCM (CALM) and the FPU predoctoral fellowship program of the Spanish Ministry of Education, Culture and Sport (reference no. FPU12/04571). It was also developed in the framework of the project "Programa de Ayudas a Grupos de Excelencia de la Región de Murcia" funded by F. Séneca (Plan Regional de Ciencia y Tecnología 2007/2010).

REFERENCES

- [1] Net2Plan – The open-source network planner [Online]. Available: <http://www.net2plan.com/> [Last accessed: April 19, 2013]
- [2] JOM – Java Optimization Modeler [Online]. Available: <http://ait.upct.es/~ppavon/jom/> [Last accessed: April 19, 2013]
- [3] R.S. Cahn: *Wide Area Network Design: Concepts and Tools for Optimization*, Morgan-Kaufmann, 1998.
- [4] M. Pióro, D. Medhi: *Routing, Flow, and Capacity Design in Communication and Computer Networks*, Morgan-Kaufmann, 2004.