The Role of Open-source Network Optimization Software in the SDN/NFV World

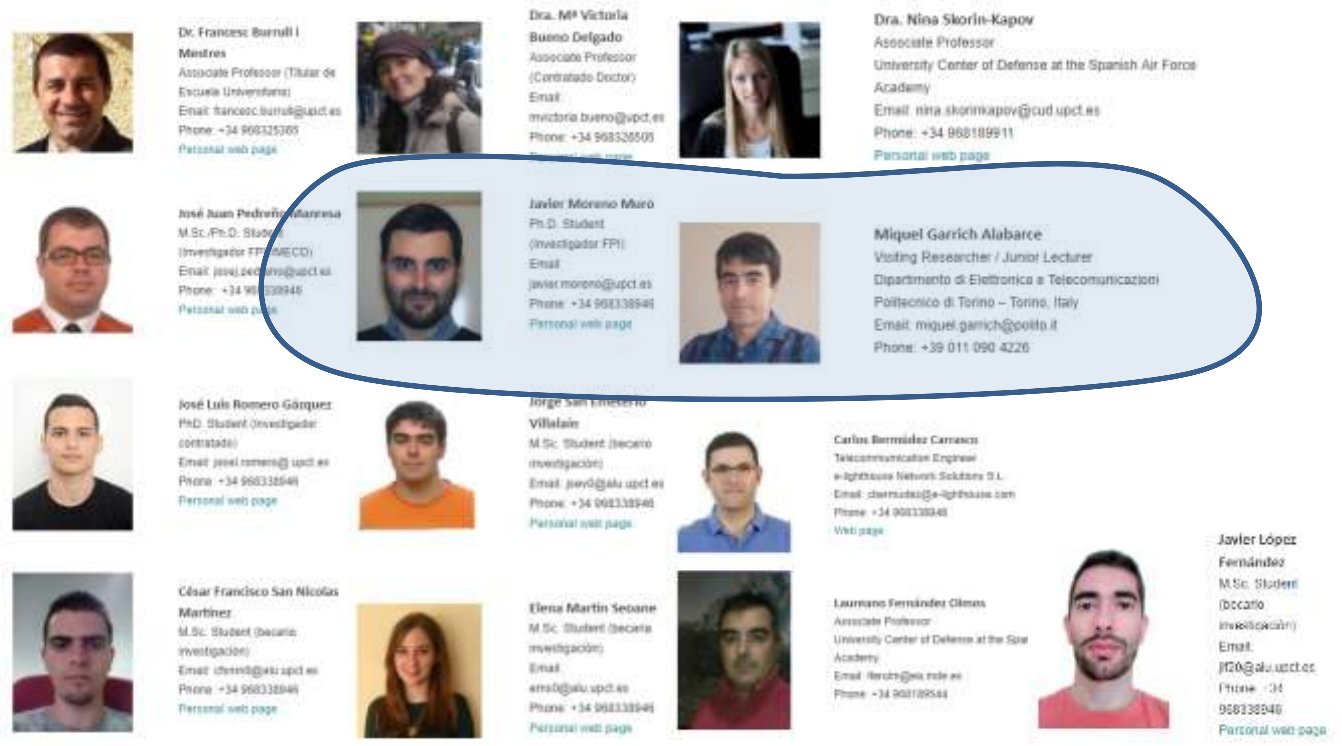
[1,2] **Pablo Pavón Mariño**
[1,3] Miquel Garrich
[1] Francisco Javier Moreno Muro

[1] Universidad Politécnica de Cartagena, (Spain)
[2] E-lighthouse Network Solutions, Spain (www.e-lighthouse.com)
[3] Optical Technologies Division, CPqD, Brazil

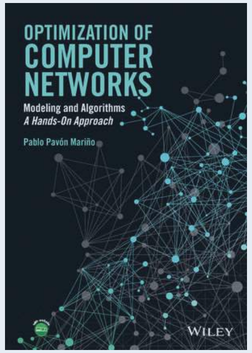
Pablo Pavón Mariño
Full Professor, Technical University of Cartagena
Head of GIRTEL research group
http://girtel.upct.es

Universidad Politécnica de Cartagena

GIRTEL



15 years track in research in network optimization



Leader of the Net2Plan open source network planning initiative

net2plan

www.net2plan.com

github.com/girtel/net2plan

Co-founder of E-lighthouse Network Solutions

Multilayer network planning software

E-LIGHTHOUSE Network Solutions

www.e-lighthouse.com


Networks 2018
18th International Telecommunications Networks
Strategy and Planning Symposium
10-12, September 2018, Los Alcázares (Spain)
networks2018.upct.es

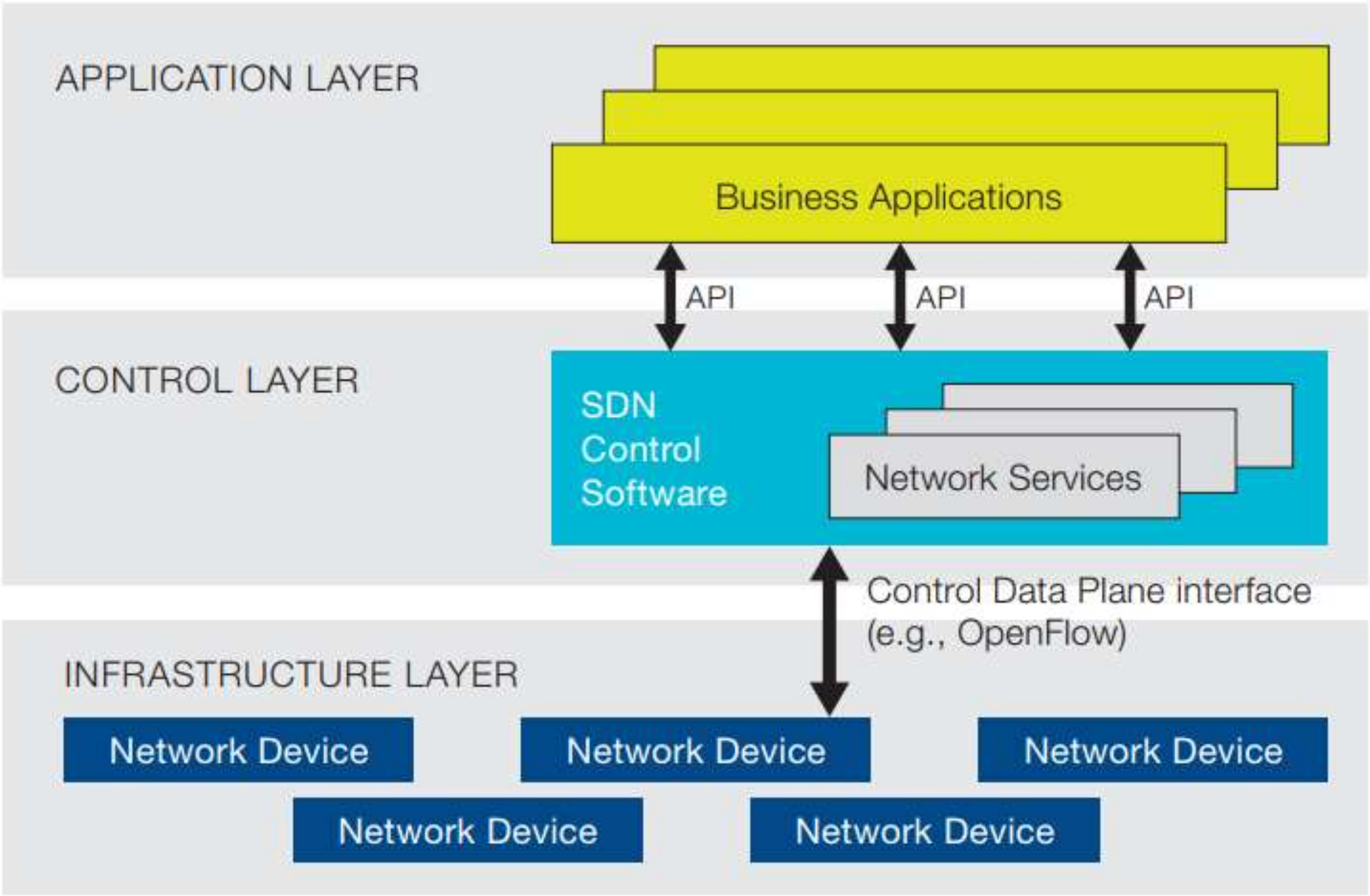
Agenda

1. Introduction
2. Use cases
3. Theoretical limits, heuristics, solvers
4. Network optimization software. Net2Plan
5. Wrap up

Agenda

1. **Introduction**
2. Use cases
3. Theoretical limits, heuristics, solvers
4. Network optimization software. Net2Plan
5. Wrap up

SDN / NFV introduction

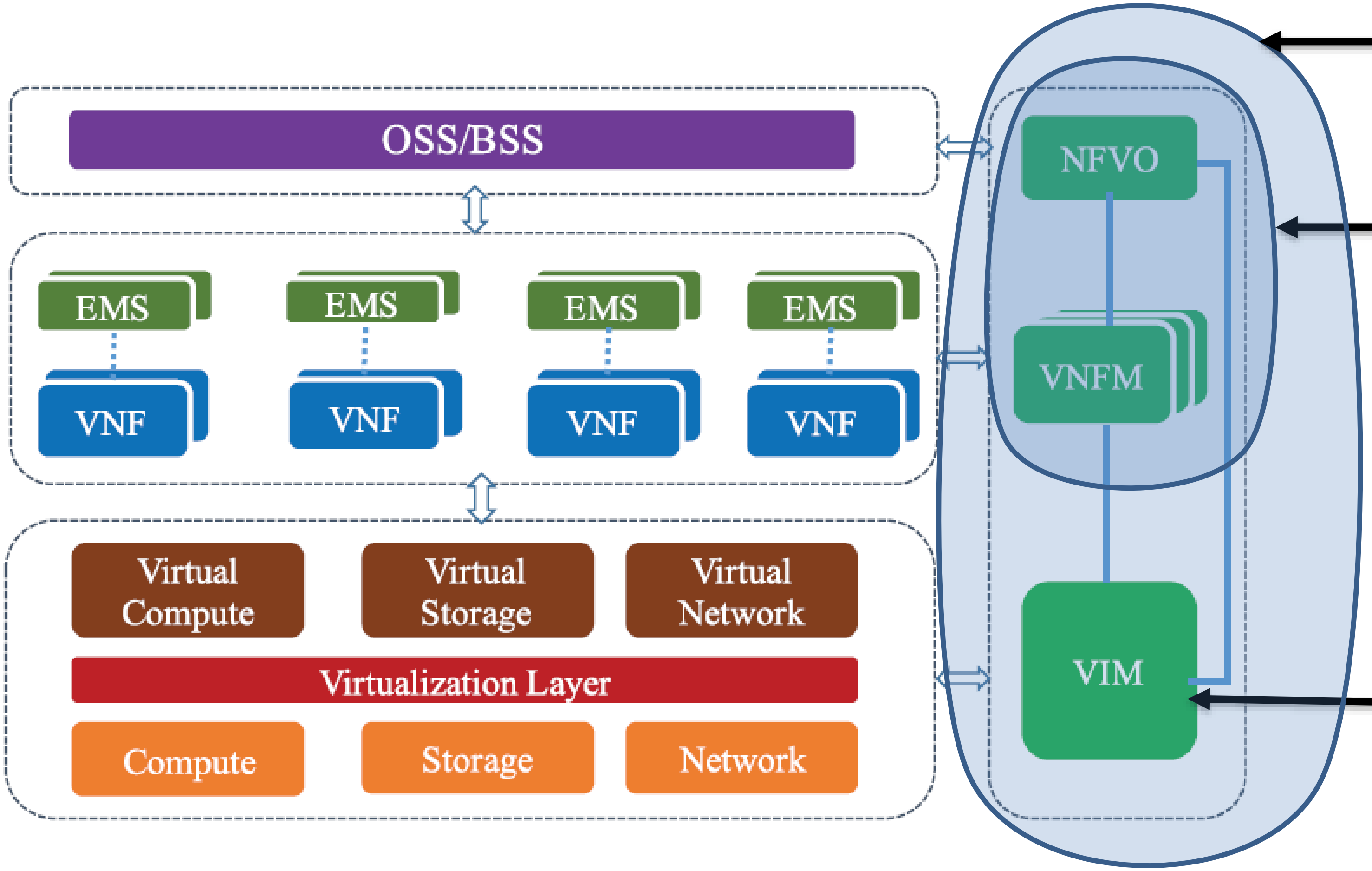


OaaS
Optimization-as-a-Service

As an SDN application

Source: ONF. Software-Defined Networking: The New Norm for Networks. White Paper April 13, 2012.
[https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf]

SDN / NFV introduction



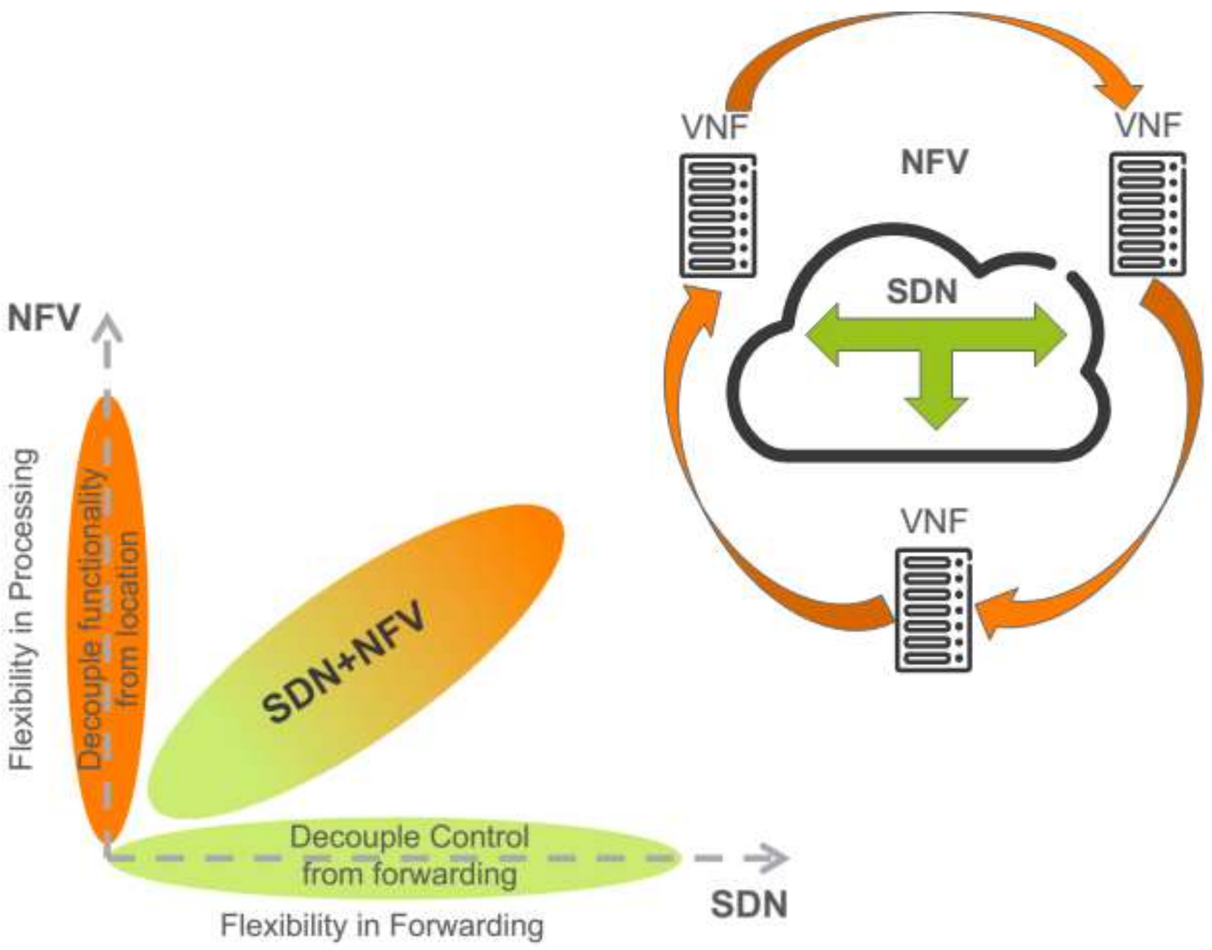
NFV MANO Architecture

E.g. Tacker (project within OpenStack)

E.g. OpenStack

ETSI NFV ISG (NFV White Paper) Oct 2012. Image source: V. G. Nguyen, et al, “SDN/NFV-based mobile packet core network architectures: a survey,” IEEE Comm. Surveys & Tutorials, 19(3), 1567-1602.

SDN / NFV introduction



Source: Ahmad Rostami (Ericsson Research)
[http://www.itc26.org/fileadmin/ITC26_files/ITC26-Tutorial-Rostami.pdf]

PROGRAMMABLE NETWORK MEANS:

- NON-MANUAL DATA ACQUISITION
- NON-MANUAL NETWORK CONTROL

WE HAVE MULTIPLE NEW USE CASES FOR OPTIMIZATION IN PRODUCTION NETWORKS...

(that so far existed just in the papers)

Agenda

1. Introduction
2. **Use cases**
3. Theoretical limits, heuristics, solvers
4. Network optimization software. Net2Plan
5. Wrap up

Use cases: optimization in practical problems

Use cases:

1. Single VNF (VM), single OpenStack
   - NOVA: Allocating VMs in computing resources, vertical scaling
2. VNF service chain, single OpenStack
   - HEAT: Allocating NSs, optimizing horizontal autoscaling
   - WATCHER: A project centralizing optimization decisions in an OpenStack
3. Network orchestration
   - Optimization as a Service (OaaS)
4. Network service across multiple OpenStacks connected in a network
   - OSM, TACKER, ONAP and others

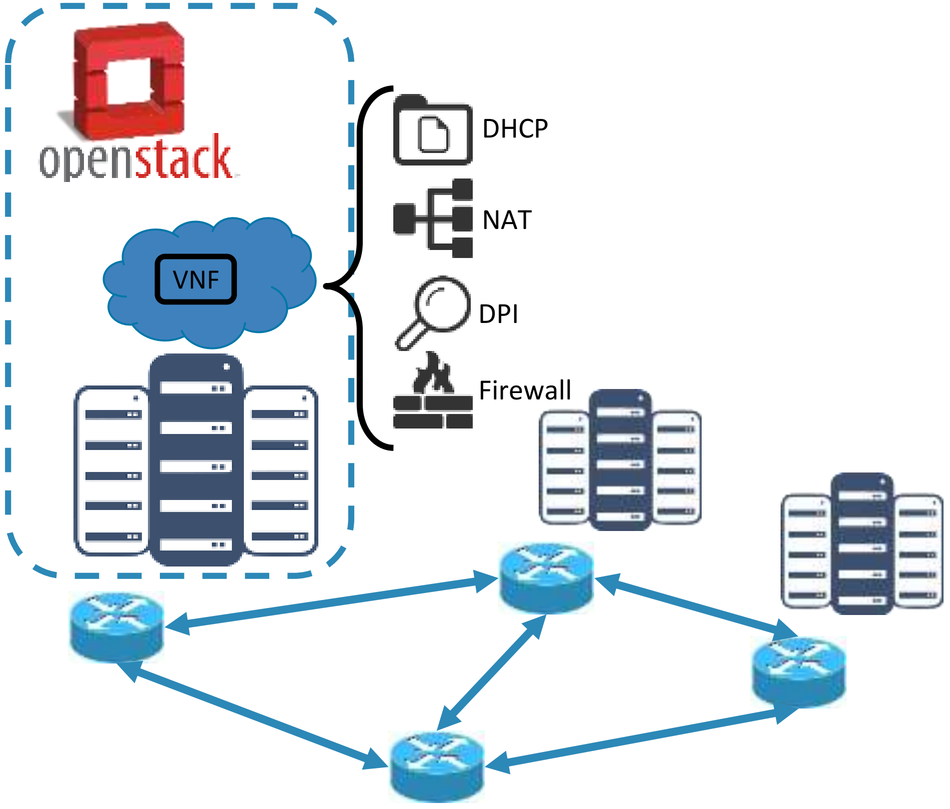
USE CASES: Optimization problems in SDN/NFV (1/4)

1. Single VNF (VM), single OpenStack

Problem: Allocation of computing resources to VNFs

Input:
- VMs CPU, HD, RAM requisites & EPA info
- Servers: CPU, HD, RAM occup. & EPA info

Output:
- Allocation of VMs in the servers



NOVA

Compute Service

Wiki page

95% Adoption

7 OF 7 Maturity

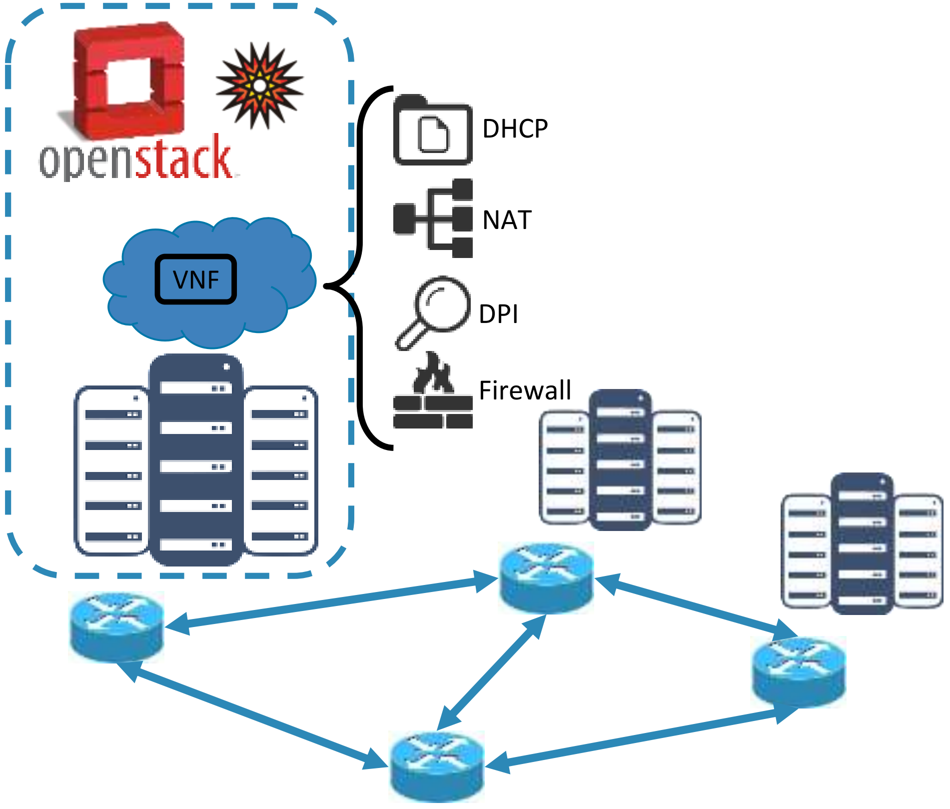
8 YRS Age

USE CASES: Optimization problems in SDN/NFV (1/4)

1. Single VNF (VM), single OpenStack

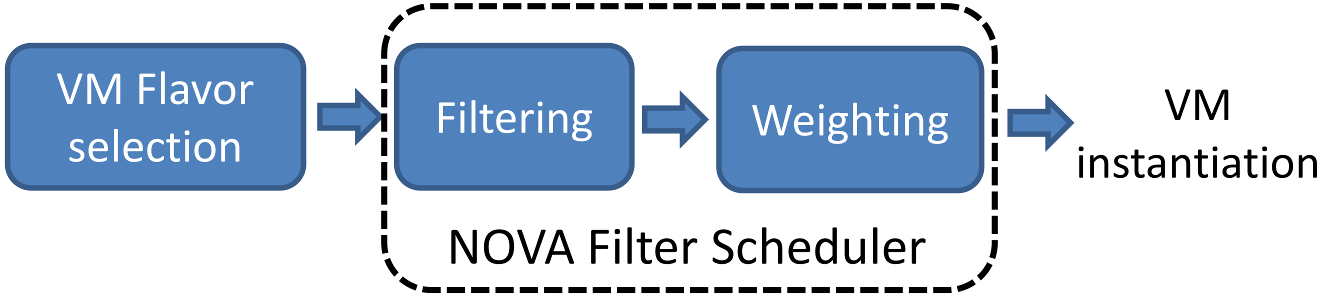
VM requisites: OpenStack flavors define the compute, memory, and storage capacity of nova computing instances.

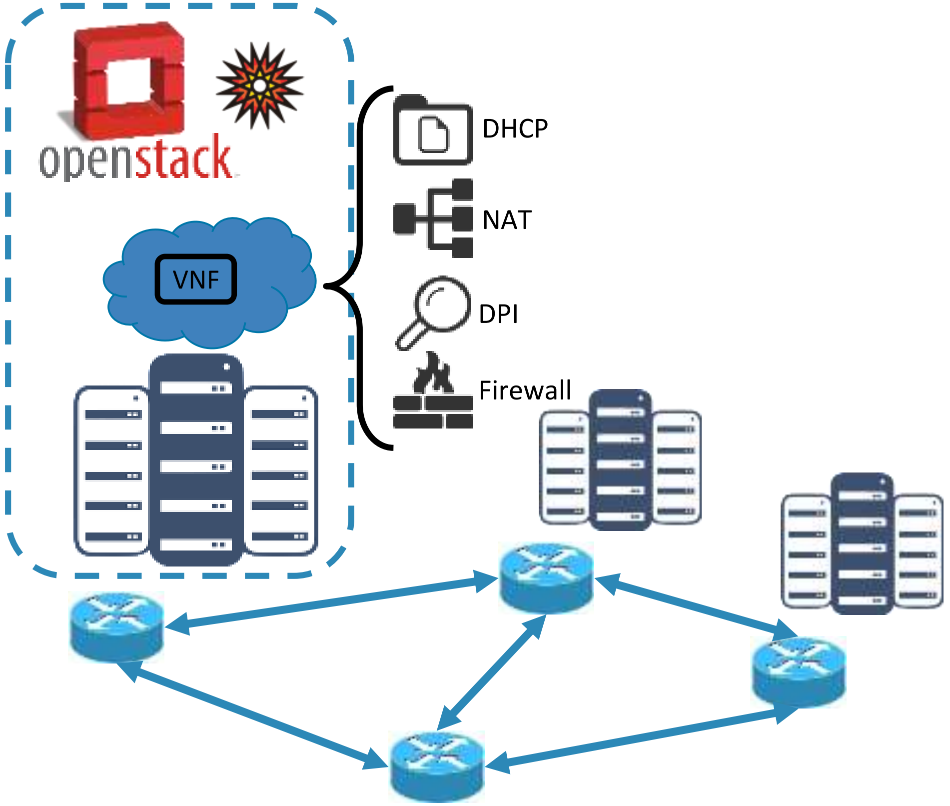
```
$ openstack flavor list
+-----+-----------+-------+------+-----------+-------+-----------+
| ID  | Name      |   RAM | Disk | Ephemeral | VCPUs | Is_Public |
+-----+-----------+-------+------+-----------+-------+-----------+
| 1   | m1.tiny   |   512 |    1 |         0 |     1 | True      |
| 2   | m1.small  |  2048 |   20 |         0 |     1 | True      |
| 3   | m1.medium |  4096 |   40 |         0 |     2 | True      |
| 4   | m1.large  |  8192 |   80 |         0 |     4 | True      |
| 5   | m1.xlarge | 16384 |  160 |         0 |     8 | True      |
+-----+-----------+-------+------+-----------+-------+-----------+
```



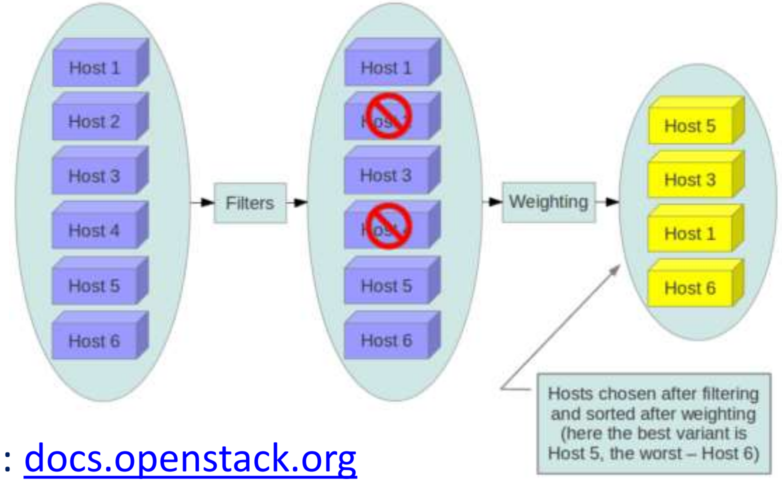
USE CASES: Optimization problems in SDN/NFV (1/4)

1. Single VNF (VM), single OpenStack





OpenStack mechanism for tuning the decision.
Programmable filters & weighters



Source: docs.openstack.org

Make your allocation algorithm and implement it as a new filter in OpenStack

USE CASES: Optimization problems in SDN/NFV (1/4)

1. Single VNF (VM), single OpenStack

Make your scheduler… things to play with

- **Availability zones** (NOVA and NEUTRON). A host can be tagged to be in one availability zone. E.g. Those connected to the same master power supply.

More info:

- https://docs.openstack.org/nova/pike/user/aggregates.html
- https://docs.openstack.org/newton/networking-guide/config-az.html

Use the information for allocations with enhanced availability

USE CASES: Optimization problems in SDN/NFV (1/4)

1. Single VNF (VM), single OpenStack

Make your scheduler... things to play with

NUMA topology (visible through the hypervisor)

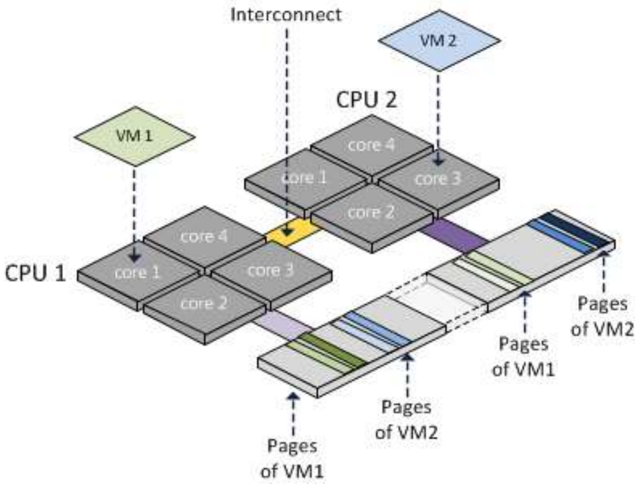
The system memory is divided into cells or nodes that are associated with a particular CPU.

CPU pinning

An instance's vCPU is assigned (pinned) to a particular host CPU.

EPA. Enhanced Platform Awareness

TOSCA VNFD* templates allow specifying VNF requirements such as NUMA topology, SR-IOV, Huge pages and CPU pinning.



Source:
https://docs.openstack.org/nova/pike/admin/cpu-topologies.html
https://www.stratoscale.com/blog/openstack/cpu-pinning-and-numa-awareness/
https://docs.openstack.org/tacker/latest/user/enhanced_placement_awareness_usage_guide.html

Optimized allocations: latency & performance

USE CASES: Optimization problems in SDN/NFV (1/4)

1. Single VNF (VM), single OpenStack

M. Scharf, M. Stein, T. Voith, and V. Hilt, "**Network-aware instance scheduling in OpenStack**," IEEE International Conference on Computer Communication and Networks (ICCCN), 2015:

- **Extension of the OpenStack scheduler** that enables a network-aware placement of instances by taking into account bandwidth constraints between nodes and exterior
- External resource tracker can monitor the allocation of bandwidth

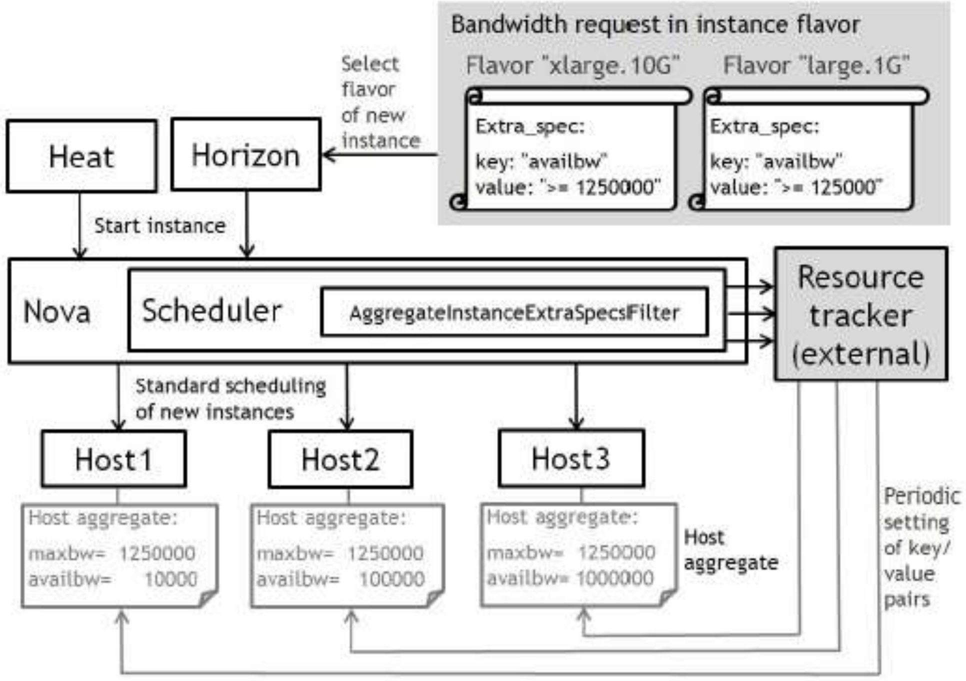
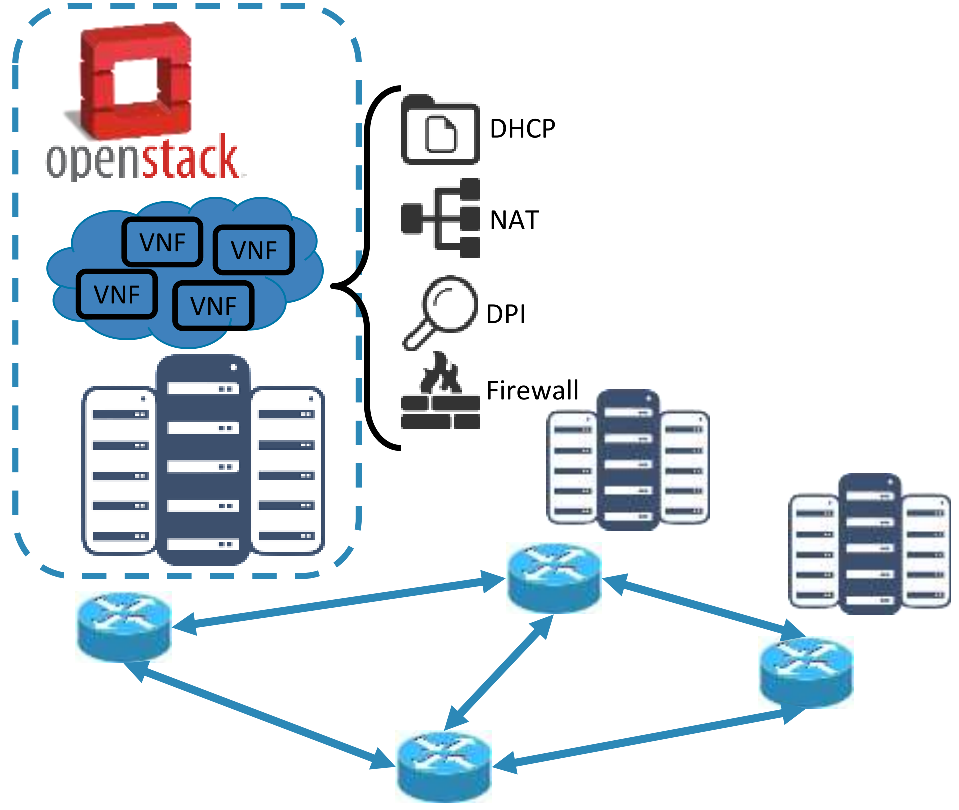


Fig. 4: Design of the host bandwidth awareness prototype

USE CASES: Optimization problems in SDN/NFV (2/4)

2. VNF service chain, single OpenStack



Problem: Allocation of network service

Input:

- Template (e.g. AWS-style, HOT) with set of VNFs, virtual connections and metadata

Output:

- Allocation of VMs in the servers
- Configuration of the virtual links between them

Heat: orchestrates the allocation of **network services** defined in templates (HOT templates), to underlying computing & networking infrastructure

HEAT

Orchestration

Wiki page

67% Adoption

4 of 7 Maturity

5 yrs Age

USE CASES: Optimization problems in SDN/NFV (2/4)

2. VNF service chain, single OpenStack

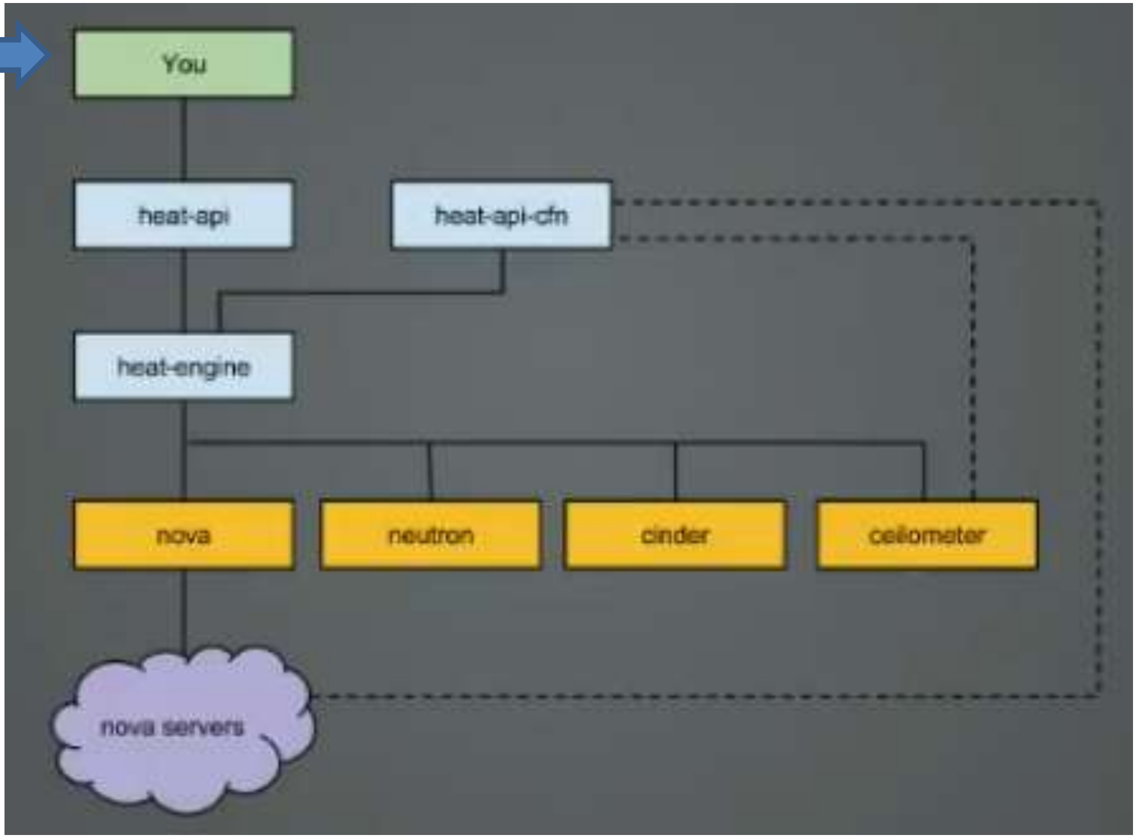
Input: HOT template

```
heat_template_version: 2015-04-30

description: Simple template to deploy a single compute instance

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      key_name: my_key
      image: ubuntu-trusty-x86_64
      flavor: m1.small
```

Heat Orchestration Template (HOT) structure

- Version
- Description
- Parameters
- (Stack of) Resources
- Output

Heat architecture components



Source: L. Kellog-Stedman, "Deploying with heat", Sept. 2014.

USE CASES: Optimization problems in SDN/NFV (2/4)

2. VNF service chain, single OpenStack

Heat: we can define an **Autoscaling Group** to create a desired count of similar resources (defined with the resource property in HOT format).

```
heat_template_version: 2015-04-30
...
resources:
  ...
  the_resource:
    type: OS::Heat::AutoScalingGroup
    properties:
      cooldown: Integer
      desired_capacity: Integer
      max_size: Integer
      min_size: Integer
      resource: {...}
      rolling_updates: {"pause_time": Number, "max_batch_size": Integer, "min_in_service": Integer}
```

Dynamically tune the autoscaling thresholds to e.g. avoid unproductive short-lived re-scalings

Source: https://docs.openstack.org/heat/pike/template_guide/openstack.html#OS::Heat::AutoScalingGroup

USE CASES: Optimization problems in SDN/NFV (2/4)

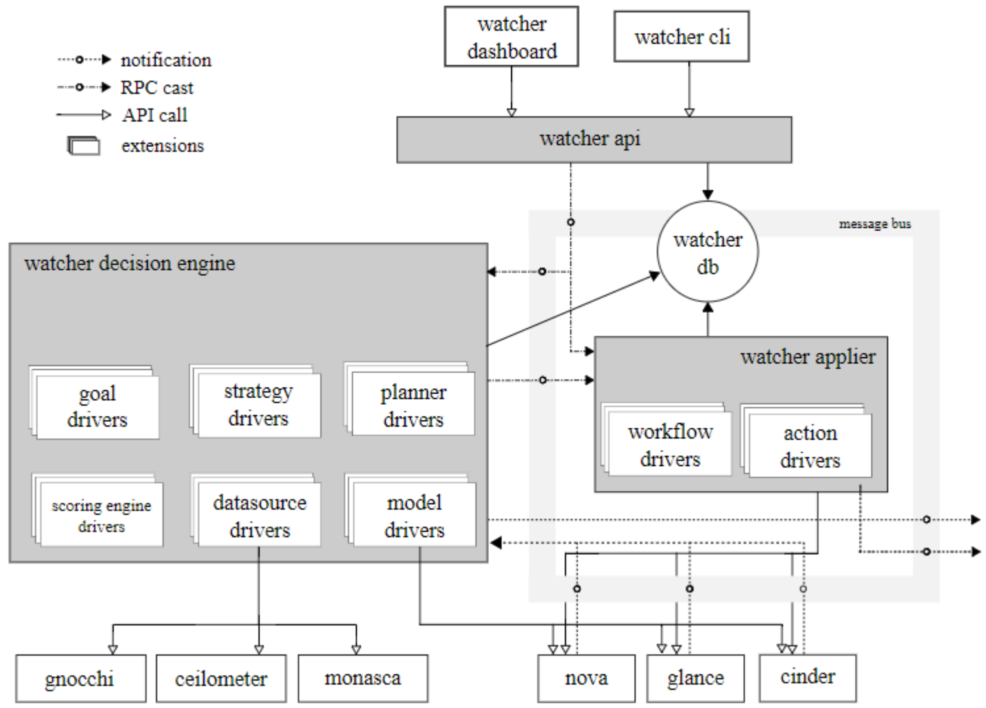
2. VNF service chain, single OpenStack

WATCHER

Optimization Service

[Wiki page]

Adoption | 1 OF 7 Maturity | 1 YRS Age



Source: https://docs.openstack.org/watcher/pike/architecture.html

- **Watcher: A project to ease a complete optimization of the OpenStack**
  - Reads monitoring information e.g. from Ceilometer service (e.g., # of vcpus, CPU utilization %, memory used)
  - Permits plugging in optimization algorithms using it

Watcher looks like a good place for capacity planning of computing/network resources in the DC & periodic application of reoptimization algorithms

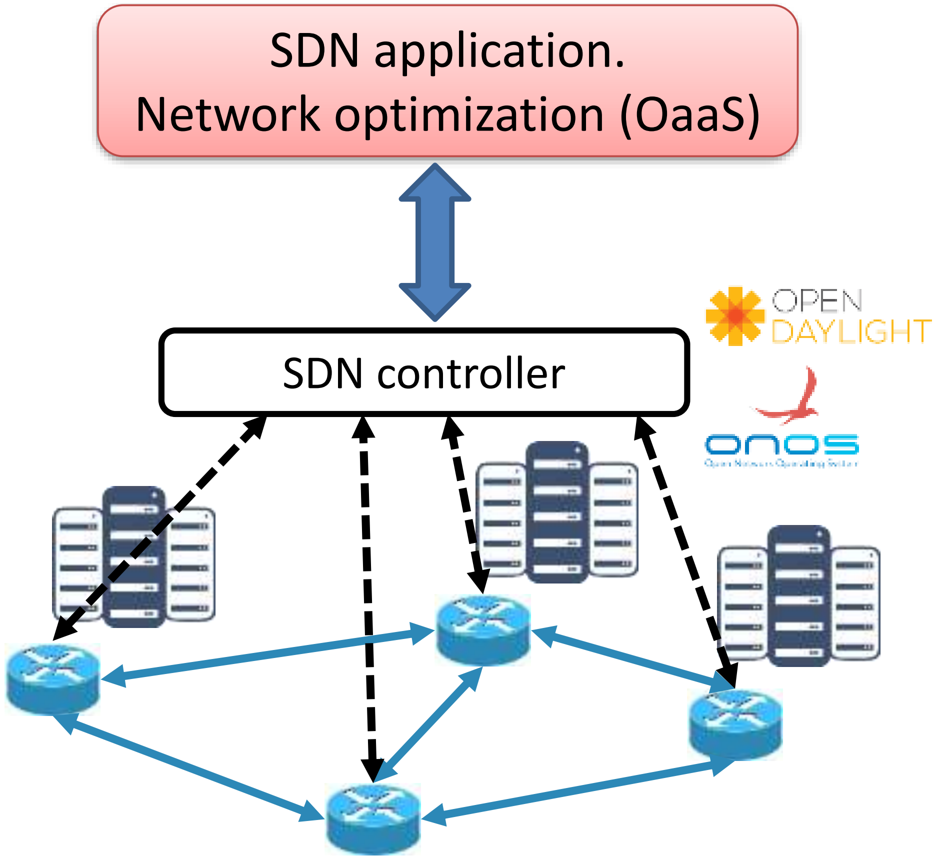
USE CASES: Optimization problems in SDN/NFV (3/4)

3. Optimization of network resources

- **Classical SDN use cases for network optimization:**
  - Periodic reoptimization
  - PCE-like path computation for provisioning

- **More options in the scope of OaaS:**
  - Capacity planning & long term network planning

Optimization-as-a-Service naturally fits as a NBI application.



USE CASES: Optimization problems in SDN/NFV (4/4)

1. Network service across multiple OpenStacks connected in a network

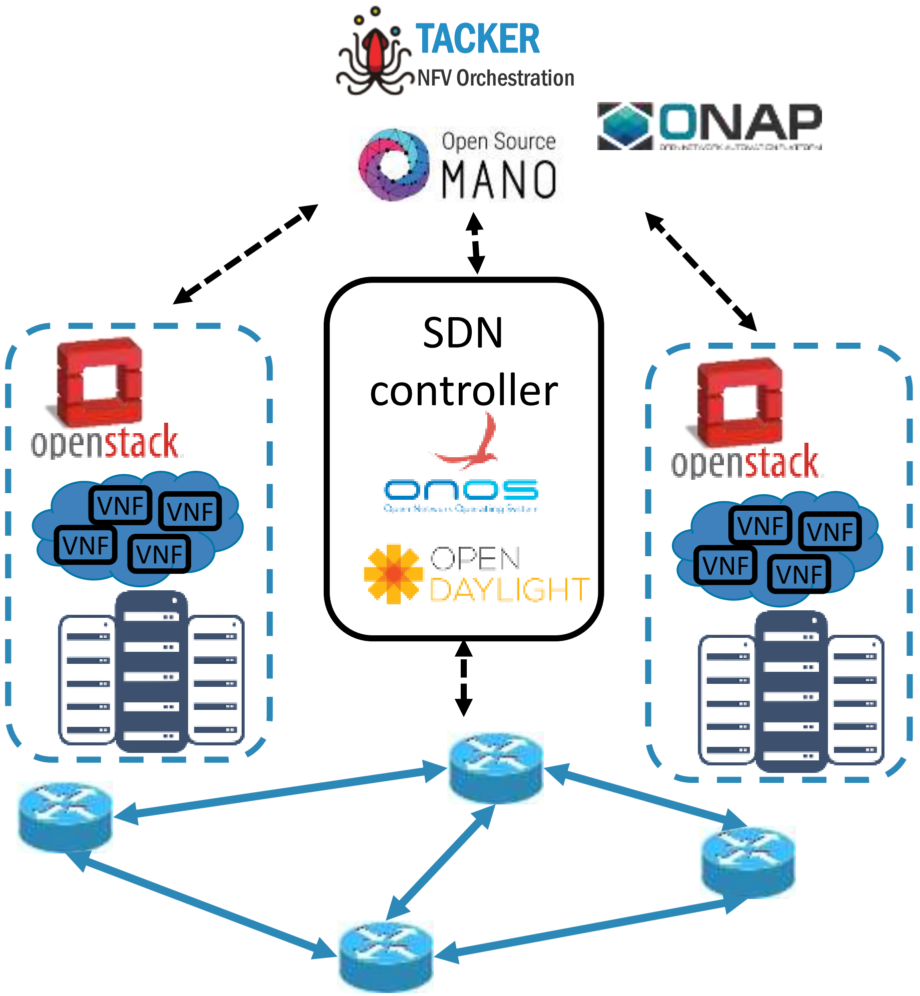
Problem: Provision network service across multiple-VIMs: **network & computing resources are JOINTLY allocated**

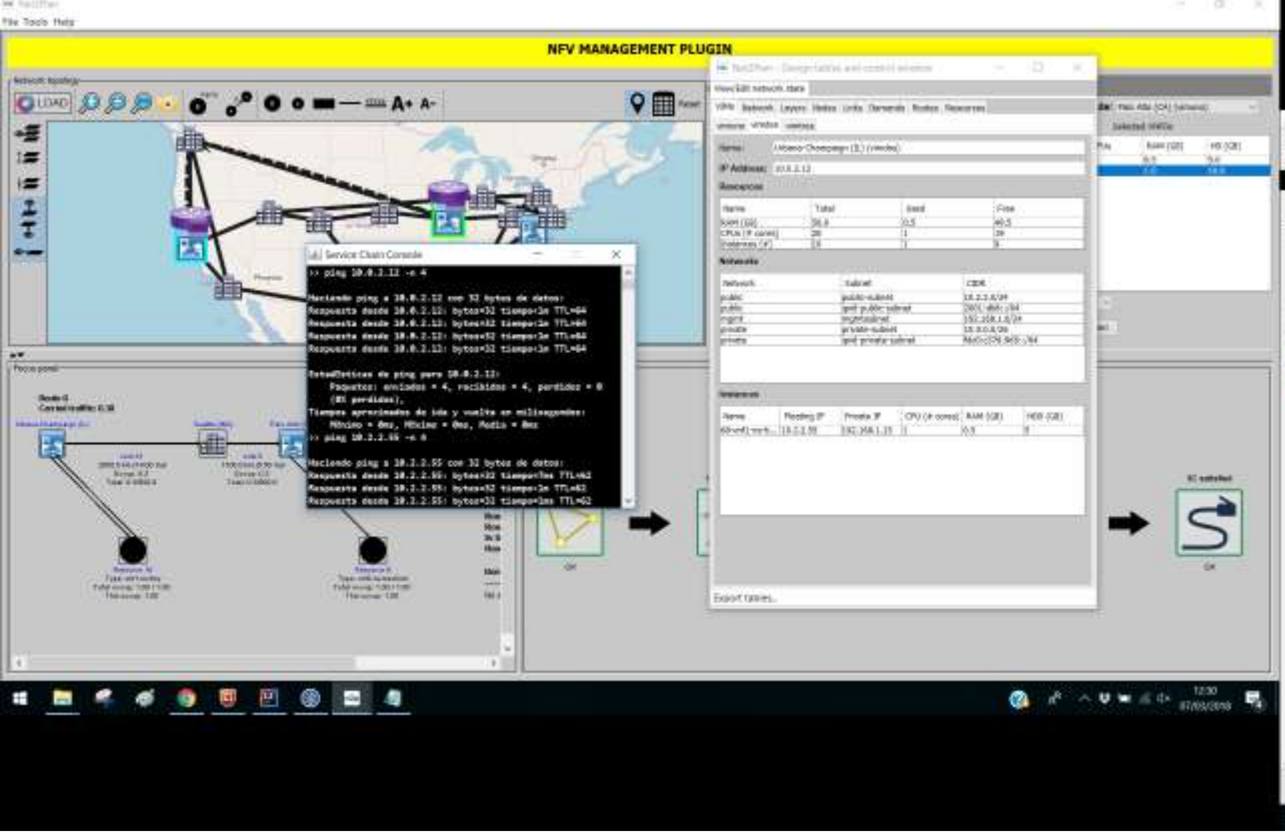
Input:

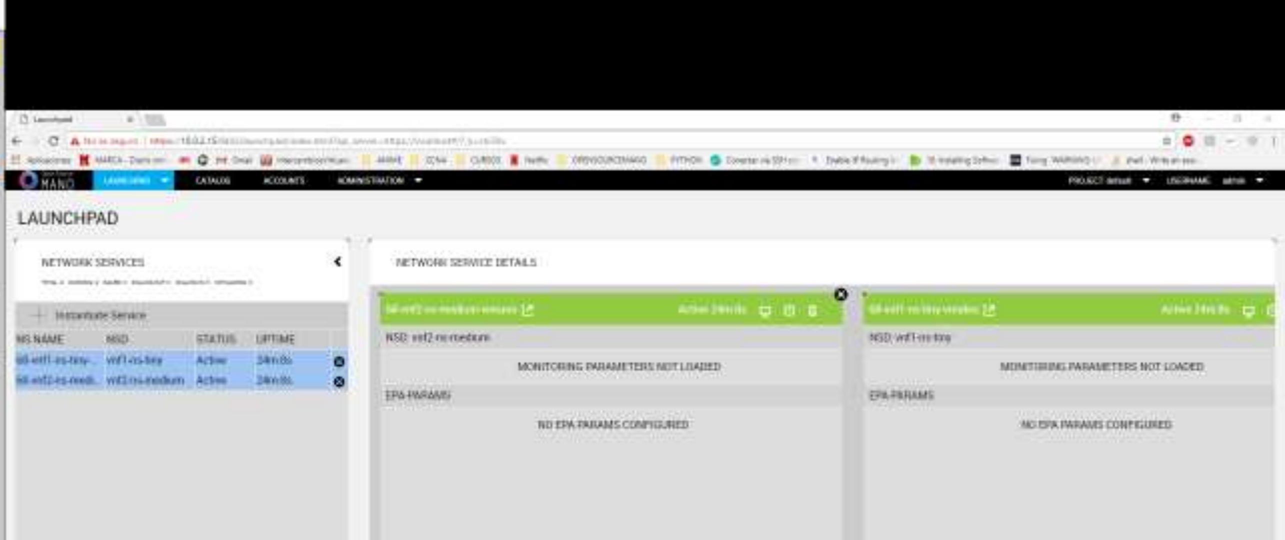
- Template (e.g. AWS-style, HOT) with set of VNFs, virtual connections and metadata

Output:

- Allocation of VMs in the servers
- Configuration of the virtual links between them
- Allocation of network resources







F.J. Moreno-Muro, C. San-Nicolas-Martinez, E. Martin-Seoane, M. Garrich, P. Pavon-Marino, O. Gonzalez de Dios, V. López, “Joint Optimal Service Chain Allocation, VNF instantiation and Metro Network Resource Management Demonstration”, OFC 2018 (SDN/NFV Demo Zone)

net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Net2Plan and SDN/NFV**
  - **OFC 2018 SDN/NFV Demo zone**. Connection of Net2Plan to **OpenStack** & **OSM** for making optimized allocations of service chains in multi-VIM environment
  - **Demonstrating joint allocation of IT (CPU, RAM, HD) and network resources**

Agenda

1. Introduction
2. Use cases
3. **Theoretical limits, heuristics, solvers**
4. Network optimization software. Net2Plan
5. Wrap up

Theoretical limits to algorithm performances

Message: There are theoretical limits to algorithm performaces
Most problems in this context are proven to be **INAPPROXIMABLE** (Assuming $\mathcal{P} \neq \mathcal{NP}$)

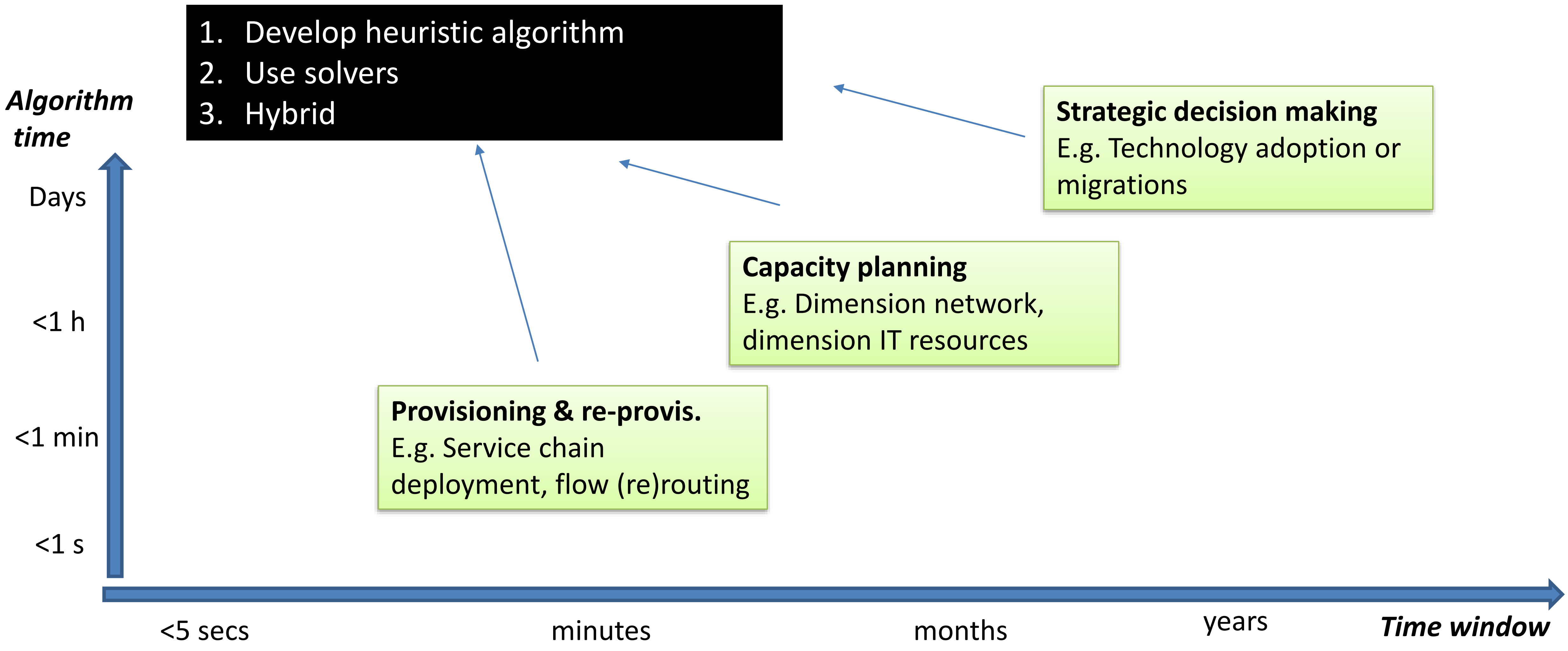
Table C.2 Complexity of some optimization problems of interest in network design.

Name	Description	Complexity
Conv	General convex programs	Polynomial ($\mathcal{P}$)
ILP	General integer linear programs	$\mathcal{NPO}$-complete
BLP	General binary linear programs	$\mathcal{NPO}$-complete
Min-kMST	k-minimum cost spanning trees	Polynomial ($\mathcal{P}$)
Min-kSP	k-minimum cost paths	Polynomial ($\mathcal{P}$)
Min-kSP	k-minimum cost paths	Polynomial ($\mathcal{P}$)
Min-Ste	Min cost multicast tree (Steiner tree)	0.55-approx., $\mathcal{APX}$-complete
Max-Clique	Maximum size clique	$\mathcal{NPO}$-complete
Min-TSP	Min cost ring	$\mathcal{NPO}$-complete
Min-NonBif	Min congestion non-bifurcated routing	2.23-approx., $\mathcal{APX}$-complete
Max-IntegralFlow	Max integral k multicommodity flow on trees	1-approx., $\mathcal{APX}$-complete
Min-NodeLocation	Min cost node location, no connectivity limit	1.4-approx., $\mathcal{APX}$-complete

Source: P. Pavon, "Optimization of computer networks. Modeling and algorithms. A hands-on approach", Wiley 2016.

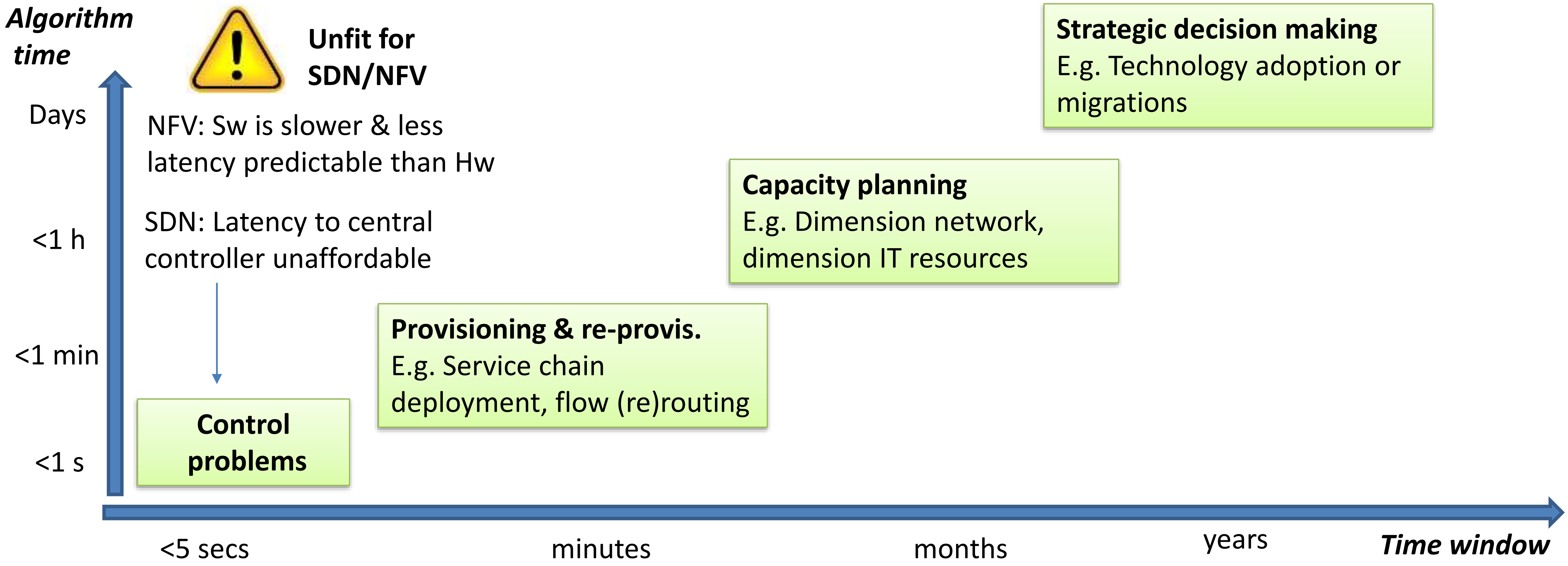
E.g. NPO-complete = No **POLYNOMIAL** algorithm exists that guarantees giving a solution at most X% worse than the optimum (for any "X")

Choose the right optimization technique



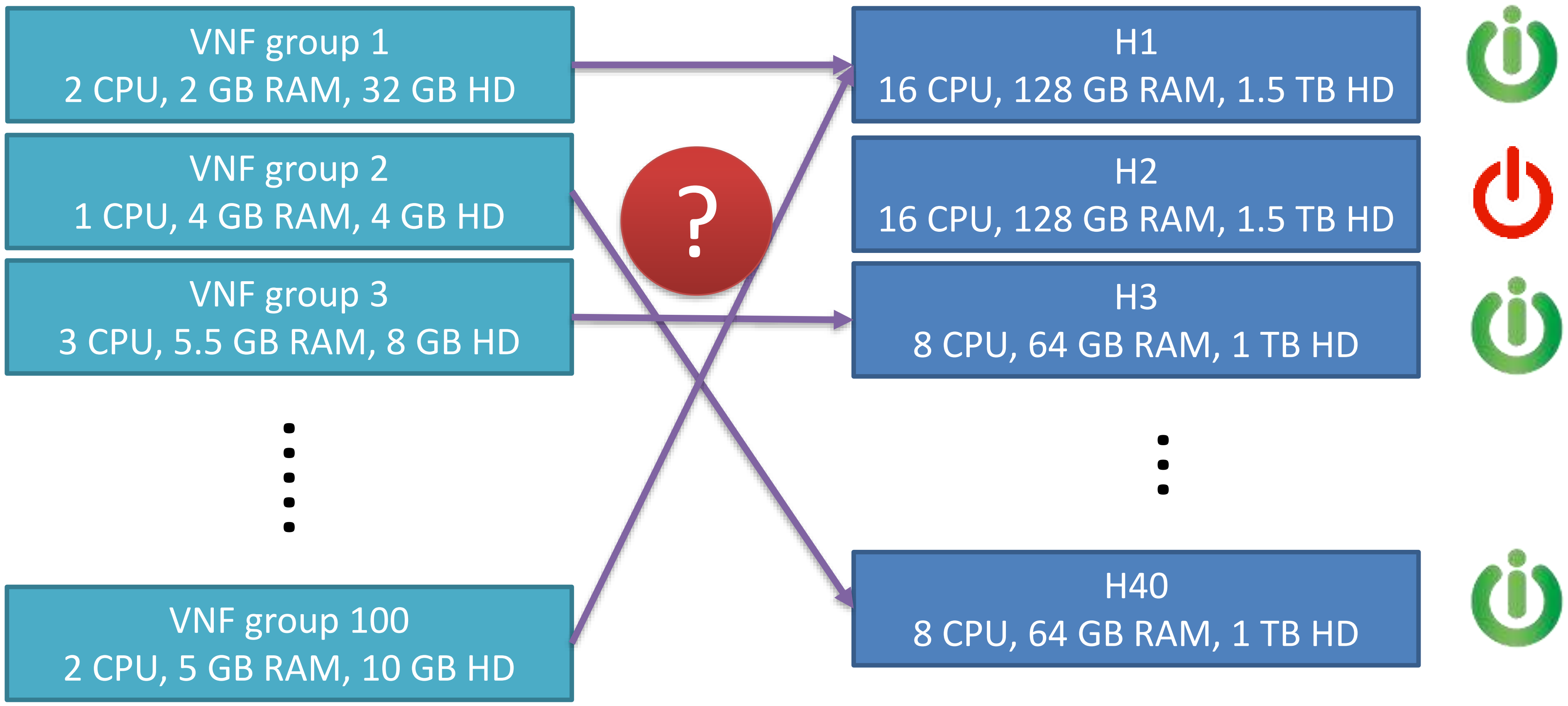
Choose the right optimization technique

Message: SDN & NFV. Choose the right optimization technique



Example

Example: Reoptimize VNF grouping assignments to servers in the DC, minimizing **energy costs** and **avoiding excessive migrations**



Example

Example: Reoptimize VNF grouping assignments to servers in the DC, minimizing **energy costs** and **avoiding excessive migrations**

$$\min \quad \sum_h a_h + \alpha \sum_{vh} c_{vh} x_{vh} \quad \text{subject to:}$$ → Minimize number of switched on hosts + weighted migration costs

$$\sum_v CPU(v) x_{vh} \leq CPU(h) a_h, \quad \forall h \in \mathcal{H}$$ → Limit to the CPUs in the hosts

$$\sum_v RAM(v) x_{vh} \leq RAM(h) a_h, \quad \forall h \in \mathcal{H}$$ → Limit to the RAM in the hosts

$$\sum_v HD(v) x_{vh} \leq HD(h) a_h, \quad \forall h \in \mathcal{H}$$ → Limit to the HD in the hosts

$$\sum_h x_{vh} = 1, \quad \forall v \in \mathcal{V}$$ → All VNF groups are allocated

$$a_h \geq x_{vh}, \forall v \in \mathcal{V}, h \in \mathcal{H}$$ → A host with at least one VNF should be switched on

Example. Heuristic vs formulation

Algorithmic options

Heuristic algorithms

- Ad-hoc developed algorithms for a purpose, no approximation guarantees
- Creative combination of different meta-heuristic techniques, e.g. genetic algorithms, tabu, greedy approaches, ...

(Mixed) Integer formulations

- The formulation is built using a MODELER, passed then to an external SOLVER
- Solver is configured with a MAXIMUM RUNNING TIME
- Fancy callbacks can be used to drive the solver

Program logic

1. Receive the input data
2. Apply the heuristic
3. Return the results to the business logic

Program logic

1. Receive the input data
2. Build the model
3. Call the external solver, and wait for answer
4. Build the response to return

Optimization tools. Solvers

Calling the solver through a external modeler

My program (Java, Python...) → Modeler (e.g. AMPL) → External solver (library)

Your problem data is here
Problem solution should arrive here

Input:
Model + Data
Output:
Solution to decision variables

Input:
Matrix of constraints
Vector of costs
Vector of lower and upper limits
Output:
Solution vector

$$\min \sum_h a_h + \alpha \sum_{vh} c_{vh} x_{vh} \quad \text{subject to:}$$
$$\sum_v CPU(v) x_{vh} \leq CPU(h) a_h, \quad \forall h \in \mathcal{H}$$
$$\sum_v RAM(v) x_{vh} \leq RAM(h) a_h, \quad \forall h \in \mathcal{H}$$
$$\sum_v HD(v) x_{vh} \leq HD(h) a_h, \quad \forall h \in \mathcal{H}$$
$$\sum_h x_{vh} = 1, \quad \forall v \in \mathcal{V}$$
$$a_h \geq x_{vh}, \forall v \in \mathcal{V}, h \in \mathcal{H}$$

Input: Model in abstract modeling language

```
## Example Two - Model with sets

set P;

param t;
param p{i in P};
param r{i in P};
param m{i in P};

var paint{i in P};

maximize profit: sum{i in P} p[i]*paint[i];
subject to time: sum{i in P} (1/r[i])*paint[i] <= t;
subject to capacity{i in P}: 0 <= paint[i] <= m[i];
```

Optimization tools. Modelers

External modelers – Specific modeling languages

- 🙂 Provide easy interaction with the most popular solvers
- 🙁 Proprietary front-ends, interaction with your program not always easy nor fast

```
## Example Two - Model with sets

set P;

param t;
param p{i in P};
param r{i in P};
param m{i in P};

var paint{i in P};

maximize profit: sum{i in P} p[i]*paint[i];
subject to time: sum{i in P} (1/r[i])*paint[i] <= t;
subject to capacity{i in P}: 0 <= paint[i] <= m[i];
```

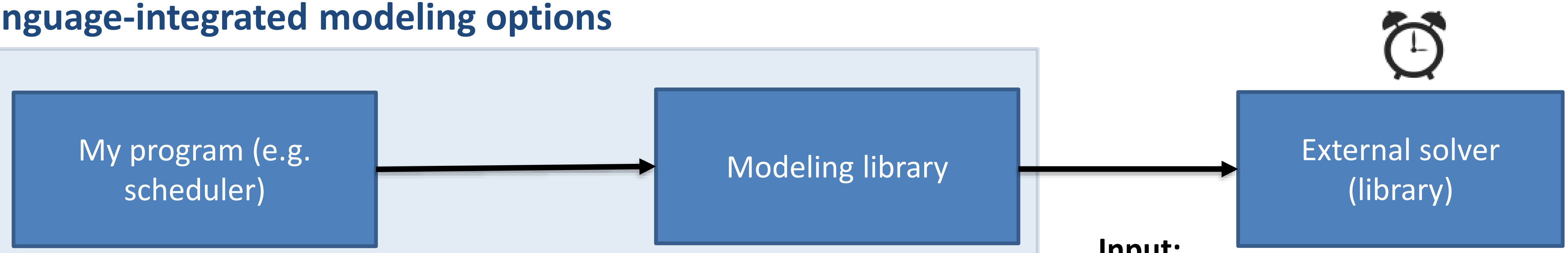
AMPL
STREAMLINED MODELING FOR REAL OPTIMIZATION

AMPL	http://www.ampl.com	Commercial
GAMS	http://www.gams.com/	Commercial
AIMMS	http://www.aimms.com	Commercial
LINGO	http://www.lindo.com	Commercial
MPL	http://www.maximalsoftware.com	Commercial
TOMLAB OPTIMIZATION	http://www.tomopt.com/	Commercial

Other front-ends (some free): CMPL (COIN-OR modeling language), GMPL (open-source AMPL clone), R (RSymphony plugin), CVX (from Stanford spinoff, for Matlab), ...

Optimization tools. Solvers

Language-integrated modeling options



$$\min \sum_h a_h + \alpha \sum_{vh} c_{vh} x_{vh} \quad \text{subject to:}$$
$$\sum_v CPU(v) x_{vh} \leq CPU(h) a_h, \quad \forall h \in \mathcal{H}$$
$$\sum_v RAM(v) x_{vh} \leq RAM(h) a_h, \quad \forall h \in \mathcal{H}$$
$$\sum_v HD(v) x_{vh} \leq HD(h) a_h, \quad \forall h \in \mathcal{H}$$
$$\sum_h x_{vh} = 1, \quad \forall v \in \mathcal{V}$$
$$a_h \geq x_{vh}, \forall v \in \mathcal{V}, h \in \mathcal{H}$$

API
Add input parameters
Add decision variables
Add constraints
Call solver
Retrieve solution & metadata

Input:
Matrix of constraints
Vector of costs
Vector of lower and upper limits
Output:
Solution vector

Optimization tools. Modelers

Language-integrated modeling options (free)

- Python
  - Pyomo. http://www.pyomo.org/
  - PuLP. https://pythonhosted.org/PuLP/
  - CyLP. https://github.com/coin-or/CyLP
  - yaposib. https://github.com/coin-or/yaposib
- C++:
  - FLOPC++. https://projects.coin-or.org/FlopC++
- Java
  - JOM (Java Optimization Modeler). http://www.net2plan.com/jom , https://github.com/girtel/JOM

Optimization tools. Solvers

Solvers

- Typically, commercial solvers incorporate libraries for calling from Java, C++, Python...
- Free solvers not always do that... rely on external modelers... or not model at all...
- Watch out! Not all the suites solve all the problem types

Name	Type
CPLEX (IBM)	Commercial
GUROBI	Commercial
XPRESS	Commercial
MOSEK	Commercial
LGO	Commercial
KNITRO	Commercial
SCIP	Free
Google Optimization tools	Free
COIN-OR (IPOPT, CLP...)	Free
GLPK	Free
SNOPT	Free
MINOS	Free
LP_SOLVE	Free
MIPCL	Free

MIP: Great grade in the tests!
(http://plato.asu.edu/ftp/milpc.html)
Around 6 times worse than commercial solvers

An example with JOM

Step 1. Initialization

Create the optimization problem object

Set the values of the input parameters (already in your program)

Set the decision variables, integer or not, minimum and maximum value

→ **Parameters, variables, and constraints can be N-dimensional arrays, dense or sparse**

$$\min \sum_h a_h + \alpha \sum_{vh} c_{vh} x_{vh} \quad \text{subject to:}$$

$$\sum_v CPU(v) x_{vh} \leq CPU(h) a_h, \quad \forall h \in \mathcal{H}$$

$$\sum_v RAM(v) x_{vh} \leq RAM(h) a_h, \quad \forall h \in \mathcal{H}$$

$$\sum_v HD(v) x_{vh} \leq HD(h) a_h, \quad \forall h \in \mathcal{H}$$

$$\sum_h x_{vh} = 1, \quad \forall v \in \mathcal{V}$$

$$a_h \geq x_{vh}, \forall v \in \mathcal{V}, h \in \mathcal{H}$$

```
/* Create the optimization problem object */
OptimizationProblem op = new OptimizationProblem();

/* Introduce in the model the values of the input parameters */
op.setInputParameter("c_vh", c_vh);
op.setInputParameter("cpu_v", cpu_v , "row");
op.setInputParameter("ram_v", ram_v , "row");
op.setInputParameter("hd_v", hd_v , "row");
op.setInputParameter("cpu_h", cpu_h , "row");
op.setInputParameter("ram_h", ram_h , "row");
op.setInputParameter("hd_h", hd_h , "row");

/* Add the decision variables to the problem */
op.addDecisionVariable("a_h", true, new int[] { 1 , H }, 0, 1);
op.addDecisionVariable("x_vh", true, new int[] { V , H }, 0, 1);
```

An example with JOM

Step 2. Set objetive function and constraints

N-dim array representation eases constraint creation

$$\min \quad \sum_h a_h + \alpha \sum_{vh} c_{vh} x_{vh} \quad \text{subject to:}$$

$$\sum_v CPU(v) x_{vh} \leq CPU(h) a_h, \quad \forall h \in \mathcal{H}$$

$$\sum_v RAM(v) x_{vh} \leq RAM(h) a_h, \quad \forall h \in \mathcal{H}$$

$$\sum_v HD(v) x_{vh} \leq HD(h) a_h, \quad \forall h \in \mathcal{H}$$

$$\sum_h x_{vh} = 1, \quad \forall v \in \mathcal{V}$$

$$a_h \geq x_{vh}, \forall v \in \mathcal{V}, h \in \mathcal{H}$$

```
/* Sets the objective function */
op.setObjectiveFunction("minimize", "sum(a_h) + 0.001 * sum(c_vh .* x_vh)");

/* Add the constraints */
op.addConstraint("cpu_v * x_vh <= cpu_h .* a_h");
op.addConstraint("ram_v * x_vh <= ram_h .* a_h");
op.addConstraint("hd_v * x_vh <= hd_h .* a_h");
op.addConstraint("sum(x_vh,2) == 1");
op.addConstraint("ones([V;1]) * a_h >= x_vh");
```

An example with JOM

Step 3. Call the solver, wait, and get the results

Solvers supported: GLPK (MILP, free), IPOPT (nonlinear, free), CPLEX (MILP, commercial), XPRESS (MILP, commercial)

Maximum solver time & solver library location are parameters

In non integer problems, gives access to Lagrange multipliers

$$\min \sum_h a_h + \alpha \sum_{vh} c_{vh} x_{vh} \quad \text{subject to:}$$

$$\sum_v CPU(v) x_{vh} \leq CPU(h) a_h, \quad \forall h \in \mathcal{H}$$

$$\sum_v RAM(v) x_{vh} \leq RAM(h) a_h, \quad \forall h \in \mathcal{H}$$

$$\sum_v HD(v) x_{vh} \leq HD(h) a_h, \quad \forall h \in \mathcal{H}$$

$$\sum_h x_{vh} = 1, \quad \forall v \in \mathcal{V}$$

$$a_h \geq x_{vh}, \forall v \in \mathcal{V}, h \in \mathcal{H}$$

```
/* Call the solver to solve the problem */
op.solve("glpk" , "maxSolverTimeInSeconds" , 5.0);
if (op.feasibleSolutionDoesNotExist())
    throw new RuntimeException ("A feasible solution does not exist");

/* Print the solution */
final double [] sol_ah = op.getPrimalSolution("a_h").to1DArray();
final double [][] sol_xvh = op.getPrimalSolution("x_vh").view2D().toArray();

/* Your code continues here */
//
//
```

Heuristic & solver combination

Idea: A heuristic iterative method guides the global optimization, solving "Mini"-MILPs in each iteration with time limits

- ☺ Heuristic in the outer loop smartly diversifies the search
- ☺ Solvers are extremely effective and fast in medium-small problems
- ⚠ We need modelers building the problem also fast! JOM can do that! ☺

Table 12.1 Case study results.

Description	1+1 protection	Shared protection	Restoration	No recovery
Total cost (K$ / year)	29506.5	28808.7	27710.7	15358.7
Link costs (K$ / year)	26046.5	25928.7	25928.7	13612.7
Transponder costs (K$ / year)	3348.0	2772.0	1674.0	1674.0
OADM costs (K$ / year)	112	108	108	72
Num. links	62	58	58	32
Num. transponders (bid)	1116	924	558	558
Num. degree 2 OADMs	0	1	1	1
Num. degree >2 OADMs	14	13	13	13
Num. GRASP iterations	11	21	21	92
Num. solver calls	1981	9421	9421	16698
Av. solver time (s)	0.241	0.194	0.194	0.052
Av. JOM modeling time (s)	1.55	0.182	0.182	0.159

SUBSECOND building + solving time for each mini-ILP*

*"Mini" means 1000s of variables and constraints

Source: P. Pavon, "Optimization of computer networks. Modeling and algorithms. A hands-on approach", Wiley 2016.

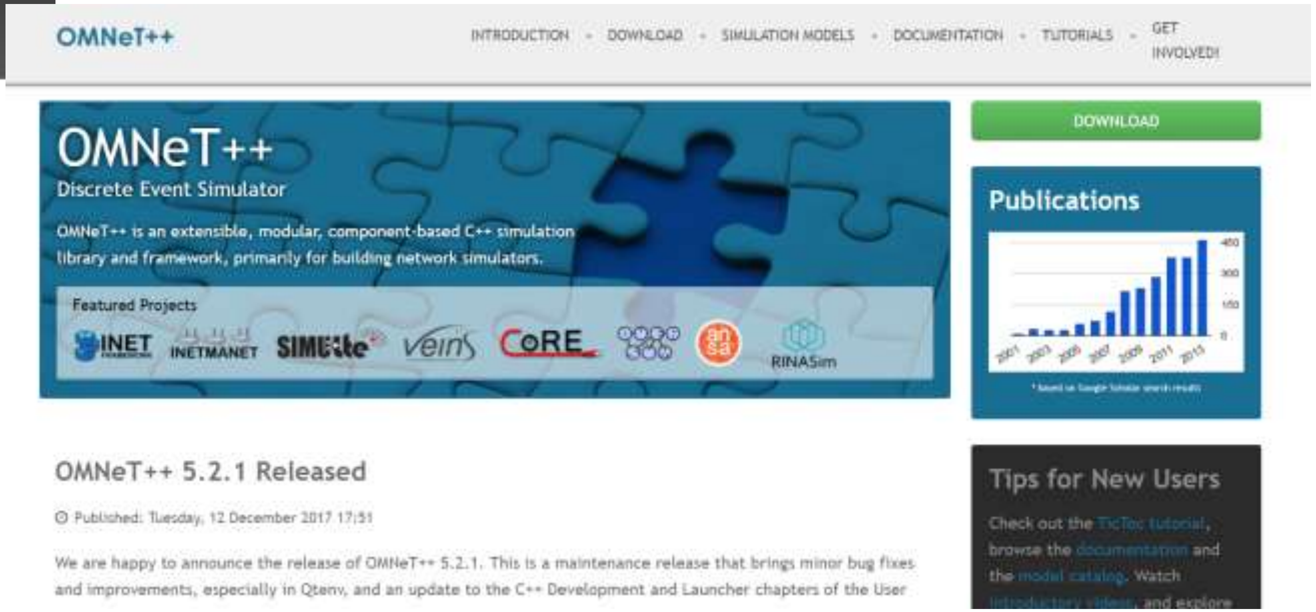
Agenda

1. Introduction
2. Use cases
3. Theoretical limits, heuristics, solvers
4. **Network optimization software. Net2Plan**
5. Wrap up

Network optimization/planning tools

What we do NOT mean with network optimization/planning tools

- Event-driven simulators, suitable for simulating network protocols at the PACKET level
- Network emulators

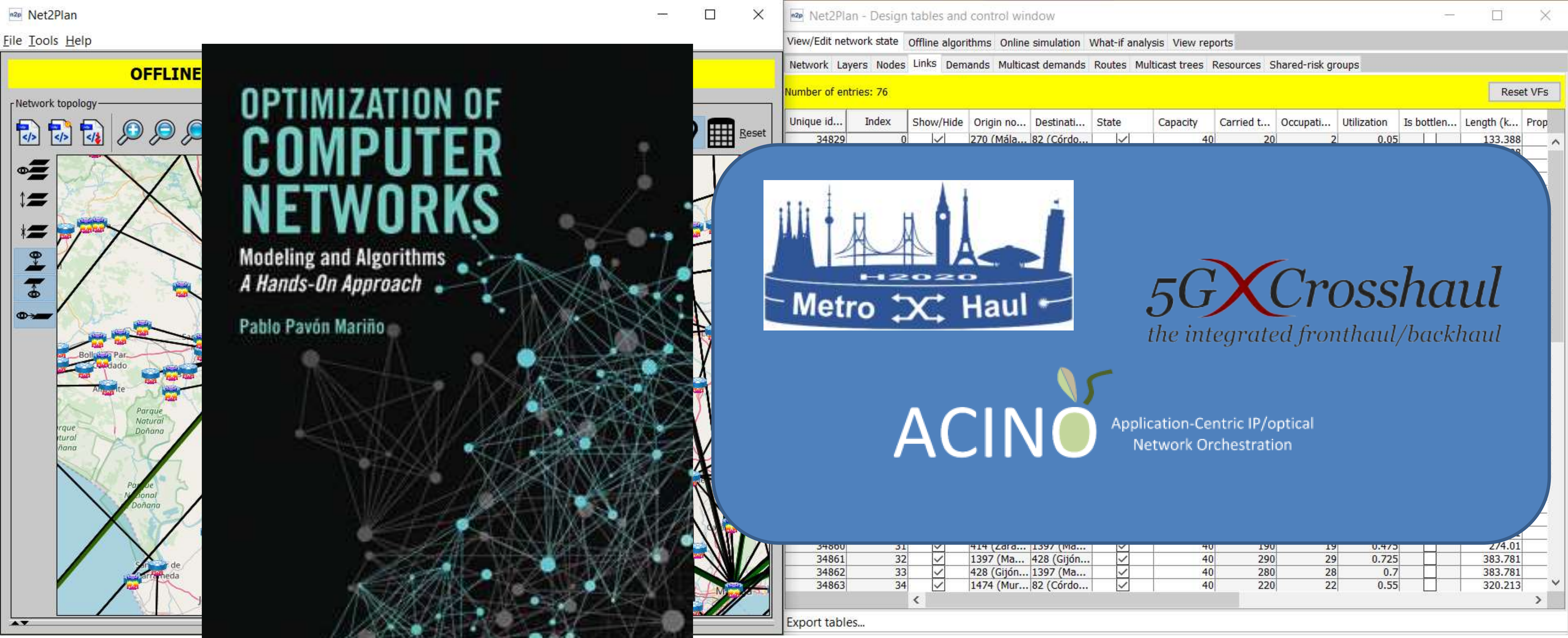


Network optimization/planning tools

What we DO mean with network optimization & planning open-source tools

- Open API for developing and plugging-in your own optimization algorithms (planning / provisioning)
- Open source, accessible in usual repositories (e.g. github)
- Some external academic and/or non-academic adoption & development maturity
- Provides CLI and/or GUI for network visualization and manipulation of the data

http://www.net2plan.com
http://github.com/girtel/net2plan

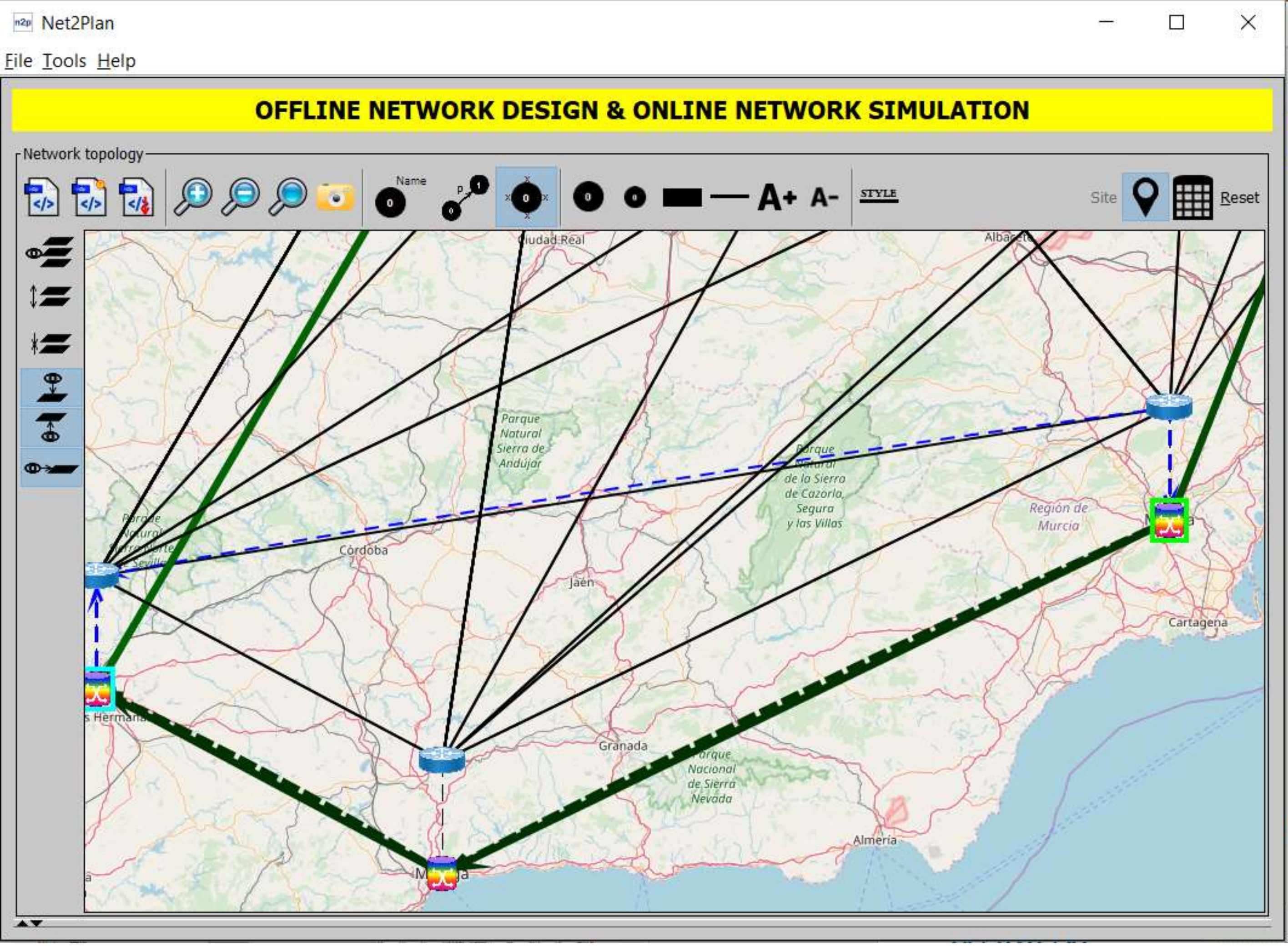


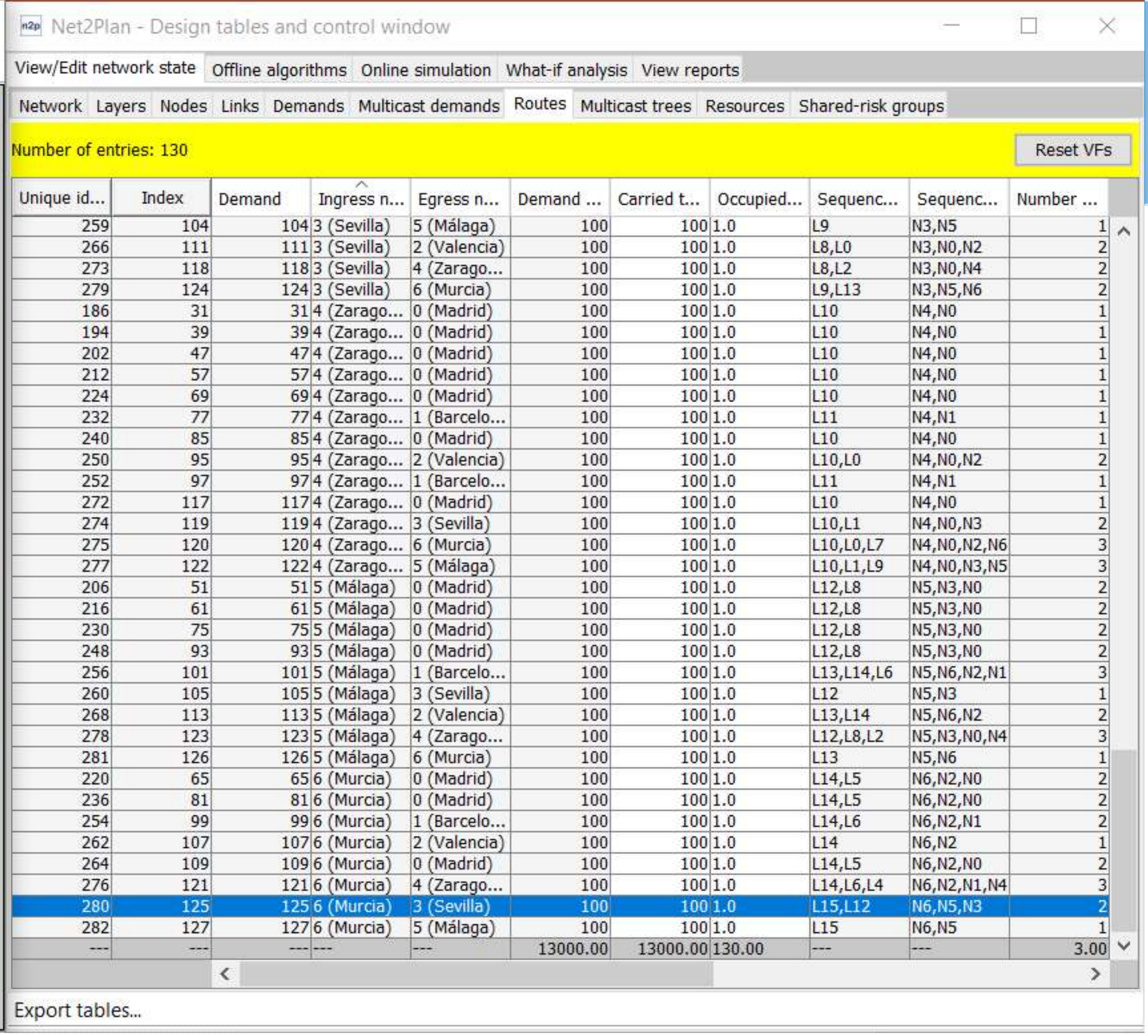
net2plan

www.net2plan.com

github.com/girtel/net2plan

- lanning tool (started in 2011)
- technology agnostic
- 0K nodes)
- State-of-the-art optimization algorithms. Open API for adding your own algorithms
- Used in a number of academic and industry projects (e.g. > 150 students/year in UPCT)





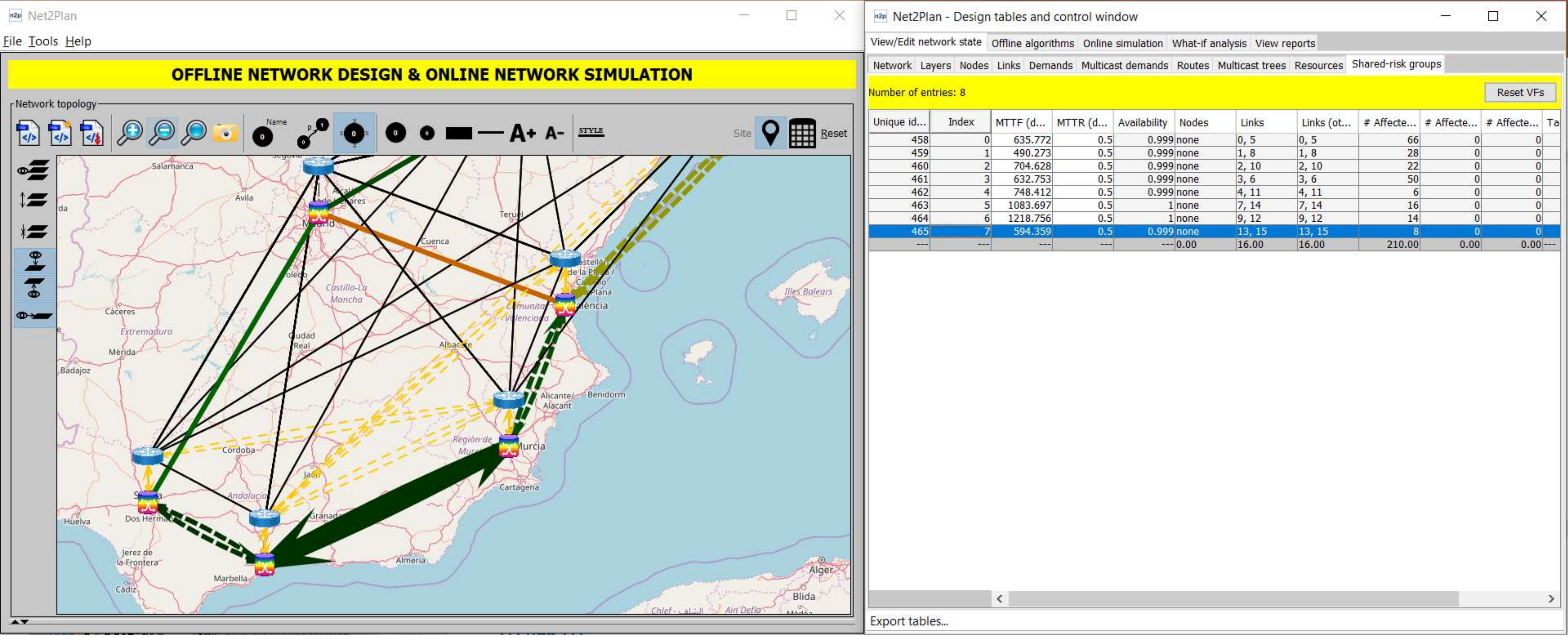
Unique id...	Index	Demand	Ingress n...	Egress n...	Demand ...	Carried t...	Occupied...	Sequenc...	Sequenc...	Number ...
259	104	104	3 (Sevilla)	5 (Málaga)	100	100	1.0	L9	N3,N5	1
266	111	111	3 (Sevilla)	2 (Valencia)	100	100	1.0	L8,L0	N3,N0,N2	2
273	118	118	3 (Sevilla)	4 (Zarago...	100	100	1.0	L8,L2	N3,N0,N4	2
279	124	124	3 (Sevilla)	6 (Murcia)	100	100	1.0	L9,L13	N3,N5,N6	2
186	31	31	4 (Zarago...	0 (Madrid)	100	100	1.0	L10	N4,N0	1
194	39	39	4 (Zarago...	0 (Madrid)	100	100	1.0	L10	N4,N0	1
202	47	47	4 (Zarago...	0 (Madrid)	100	100	1.0	L10	N4,N0	1
212	57	57	4 (Zarago...	0 (Madrid)	100	100	1.0	L10	N4,N0	1
224	69	69	4 (Zarago...	0 (Madrid)	100	100	1.0	L10	N4,N0	1
232	77	77	4 (Zarago...	1 (Barcelo...	100	100	1.0	L11	N4,N1	1
240	85	85	4 (Zarago...	0 (Madrid)	100	100	1.0	L10	N4,N0	1
250	95	95	4 (Zarago...	2 (Valencia)	100	100	1.0	L10,L0	N4,N0,N2	2
252	97	97	4 (Zarago...	1 (Barcelo...	100	100	1.0	L11	N4,N1	1
272	117	117	4 (Zarago...	0 (Madrid)	100	100	1.0	L10	N4,N0	1
274	119	119	4 (Zarago...	3 (Sevilla)	100	100	1.0	L10,L1	N4,N0,N3	2
275	120	120	4 (Zarago...	6 (Murcia)	100	100	1.0	L10,L0,L7	N4,N0,N2,N6	3
277	122	122	4 (Zarago...	5 (Málaga)	100	100	1.0	L10,L1,L9	N4,N0,N3,N5	3
206	51	51	5 (Málaga)	0 (Madrid)	100	100	1.0	L12,L8	N5,N3,N0	2
216	61	61	5 (Málaga)	0 (Madrid)	100	100	1.0	L12,L8	N5,N3,N0	2
230	75	75	5 (Málaga)	0 (Madrid)	100	100	1.0	L12,L8	N5,N3,N0	2
248	93	93	5 (Málaga)	0 (Madrid)	100	100	1.0	L12,L8	N5,N3,N0	2
256	101	101	5 (Málaga)	1 (Barcelo...	100	100	1.0	L13,L14,L6	N5,N6,N2,N1	3
260	105	105	5 (Málaga)	3 (Sevilla)	100	100	1.0	L12	N5,N3	1
268	113	113	5 (Málaga)	2 (Valencia)	100	100	1.0	L13,L14	N5,N6,N2	2
278	123	123	5 (Málaga)	4 (Zarago...	100	100	1.0	L12,L8,L2	N5,N3,N0,N4	3
281	126	126	5 (Málaga)	6 (Murcia)	100	100	1.0	L13	N5,N6	1
220	65	65	6 (Murcia)	0 (Madrid)	100	100	1.0	L14,L5	N6,N2,N0	2
236	81	81	6 (Murcia)	0 (Madrid)	100	100	1.0	L14,L5	N6,N2,N0	2
254	99	99	6 (Murcia)	1 (Barcelo...	100	100	1.0	L14,L6	N6,N2,N1	2
262	107	107	6 (Murcia)	2 (Valencia)	100	100	1.0	L14	N6,N2	1
264	109	109	6 (Murcia)	0 (Madrid)	100	100	1.0	L14,L5	N6,N2,N0	2
276	121	121	6 (Murcia)	4 (Zarago...	100	100	1.0	L14,L6,L4	N6,N2,N1,N4	3
280	125	125	6 (Murcia)	3 (Sevilla)	100	100	1.0	L15,L12	N6,N5,N3	2
282	127	127	6 (Murcia)	5 (Málaga)	100	100	1.0	L15	N6,N5	1
---	---	---	---	---	13000.00	13000.00	130.00	---	---	3.00

net2plan

www.net2plan.com

github.com/girtel/net2plan

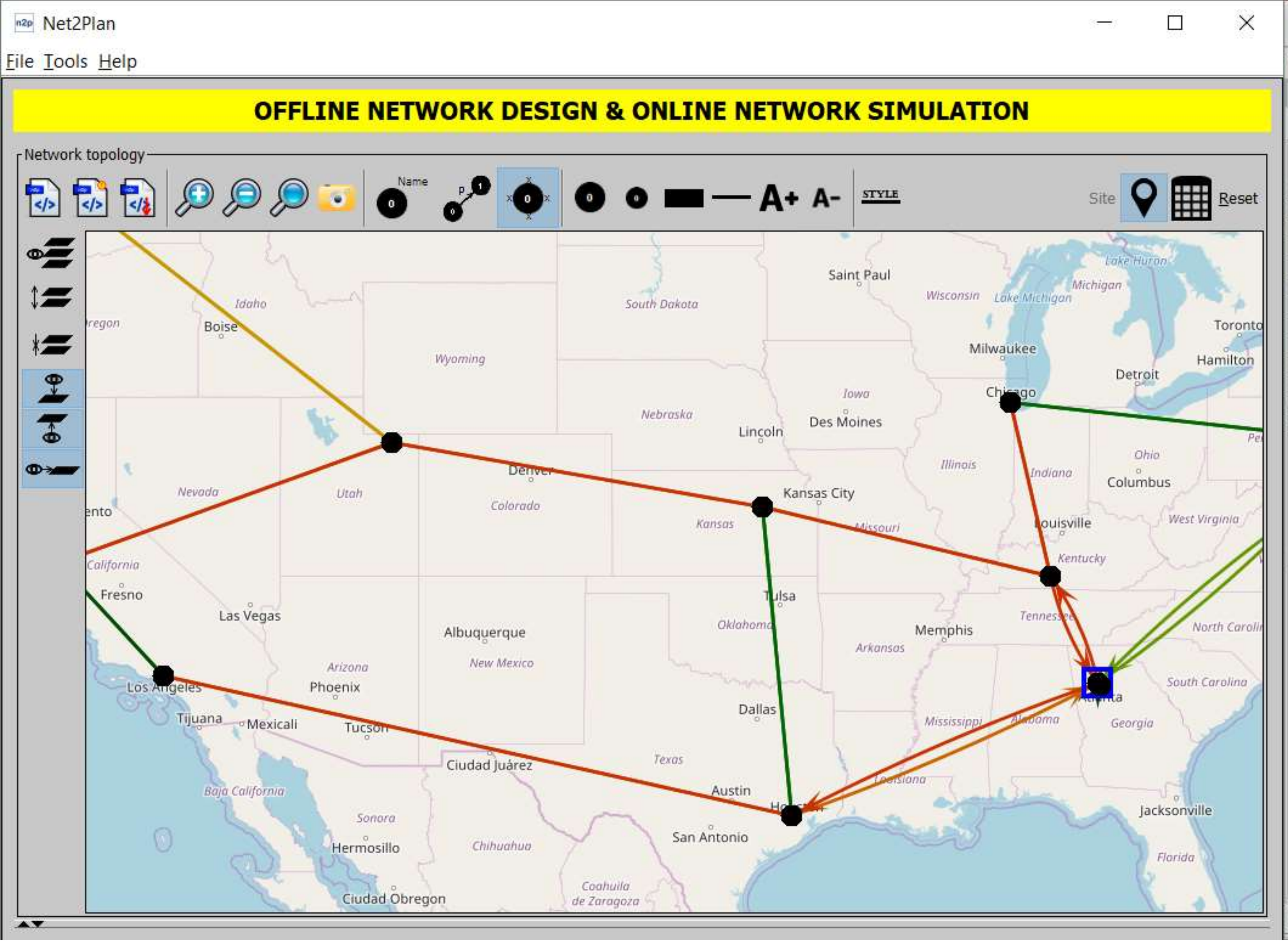
- Multilayer visualization
  - Abstract **MULTILAYER** model: Layer, Node, Link, Demand, Multicast demand, Route, SRG...
  - Tables full of **technology-agnostic statistics** (loads, utilizations, latencies...)
  - **Technology-related information** can be added as ATTRIBUTES, transparent to Net2Plan, but that algorithms can process to create technology-related behaviors

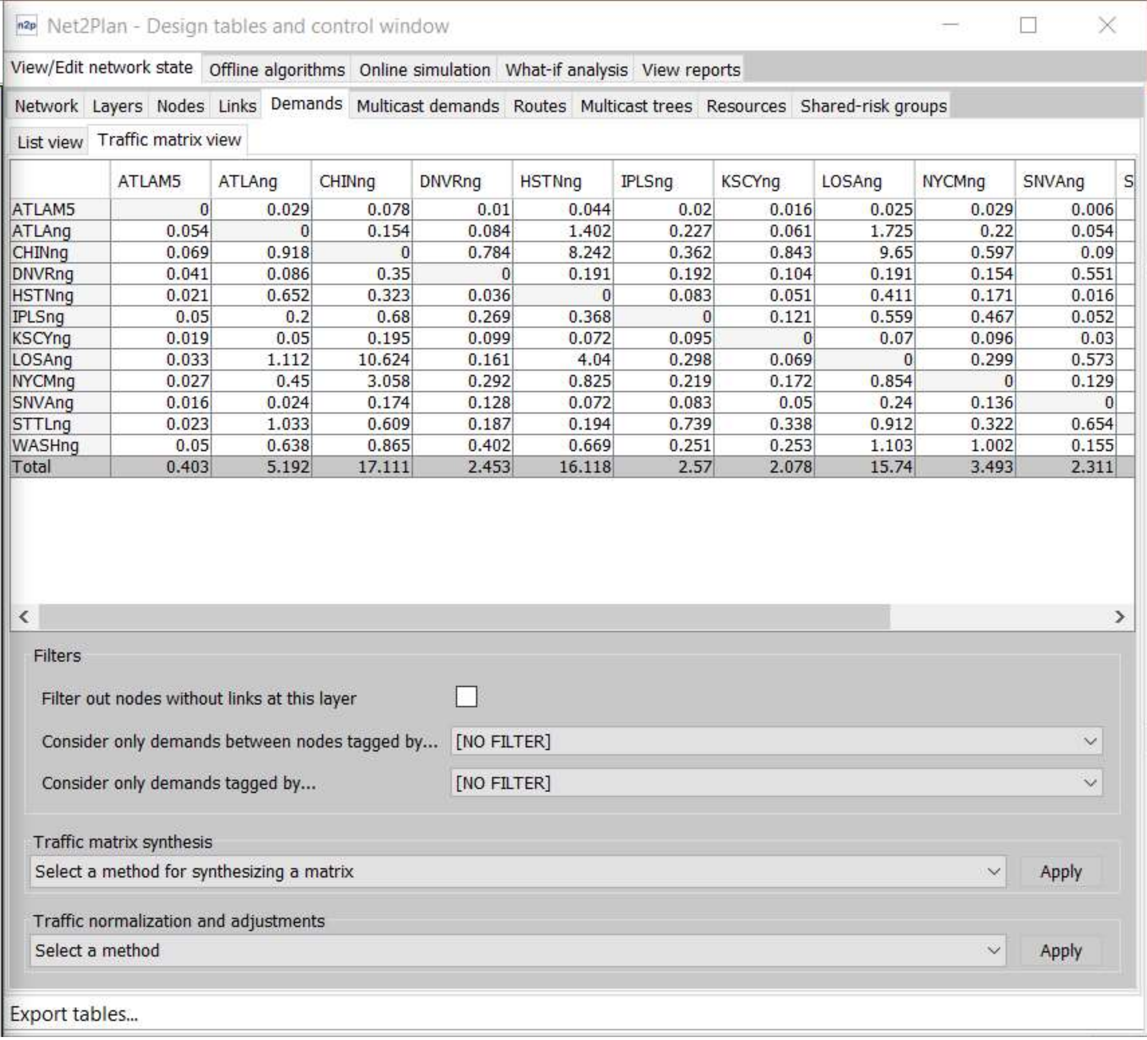


www.net2plan.com

github.com/girtel/net2plan

- **Failure model**: shared-risk-group (SRG) definition
  - SRG: arbitrary set of nodes and/or links that can simultaneously fail
  - Represents a vulnerability (e.g. duct cut)
  - Algorithms can typically try to create designs tolerant to all single-SRG failures





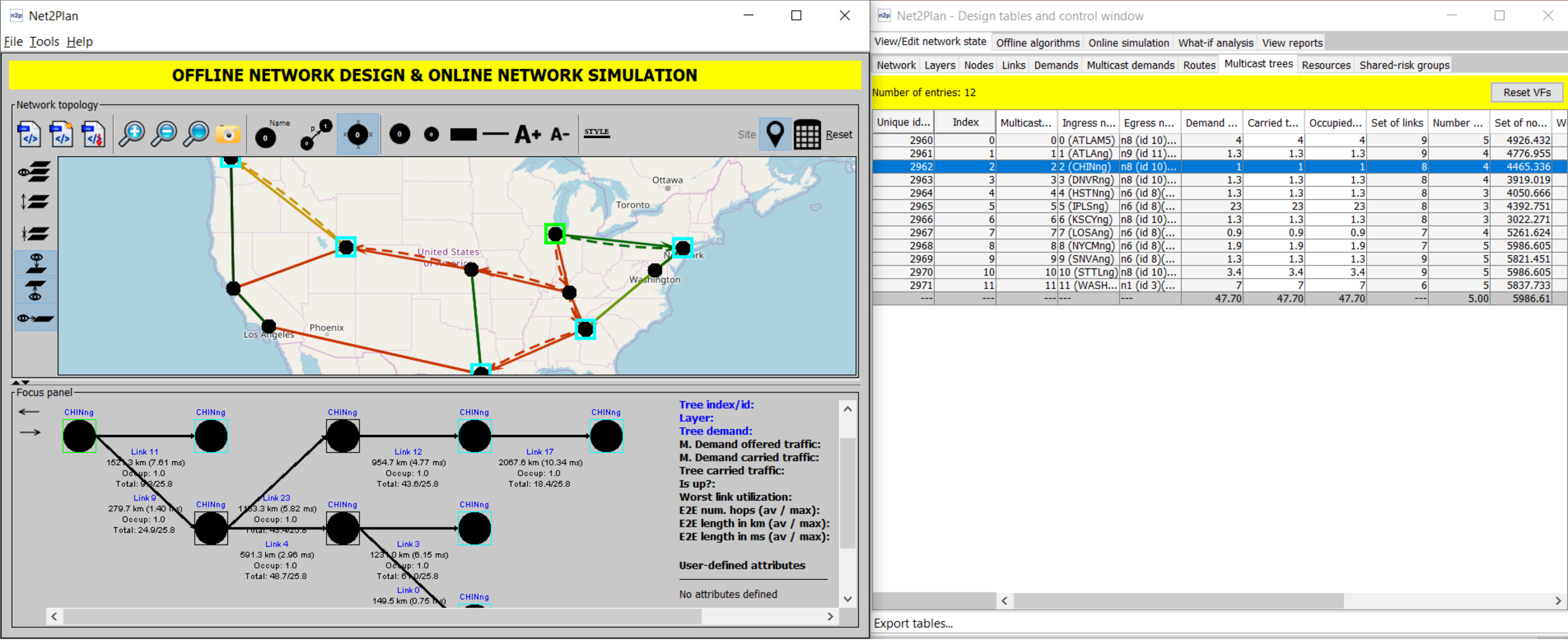
	ATLAM5	ATLAng	CHINng	DNVRng	HSTNng	IPLSng	KSCYng	LOSAng	NYCMng	SNVAng
ATLAM5	0	0.029	0.078	0.01	0.044	0.02	0.016	0.025	0.029	0.006
ATLAng	0.054	0	0.154	0.084	1.402	0.227	0.061	1.725	0.22	0.054
CHINng	0.069	0.918	0	0.784	8.242	0.362	0.843	9.65	0.597	0.09
DNVRng	0.041	0.086	0.35	0	0.191	0.192	0.104	0.191	0.154	0.551
HSTNng	0.021	0.652	0.323	0.036	0	0.083	0.051	0.411	0.171	0.016
IPLSng	0.05	0.2	0.68	0.269	0.368	0	0.121	0.559	0.467	0.052
KSCYng	0.019	0.05	0.195	0.099	0.072	0.095	0	0.07	0.096	0.03
LOSAng	0.033	1.112	10.624	0.161	4.04	0.298	0.069	0	0.299	0.573
NYCMng	0.027	0.45	3.058	0.292	0.825	0.219	0.172	0.854	0	0.129
SNVAng	0.016	0.024	0.174	0.128	0.072	0.083	0.05	0.24	0.136	0
STTLng	0.023	1.033	0.609	0.187	0.194	0.739	0.338	0.912	0.322	0.654
WASHng	0.05	0.638	0.865	0.402	0.669	0.251	0.253	1.103	1.002	0.155
Total	0.403	5.192	17.111	2.453	16.118	2.57	2.078	15.74	3.493	2.311

- **Traffic matrix synthesis, manipulation, normalization**

net2plan

www.net2plan.com

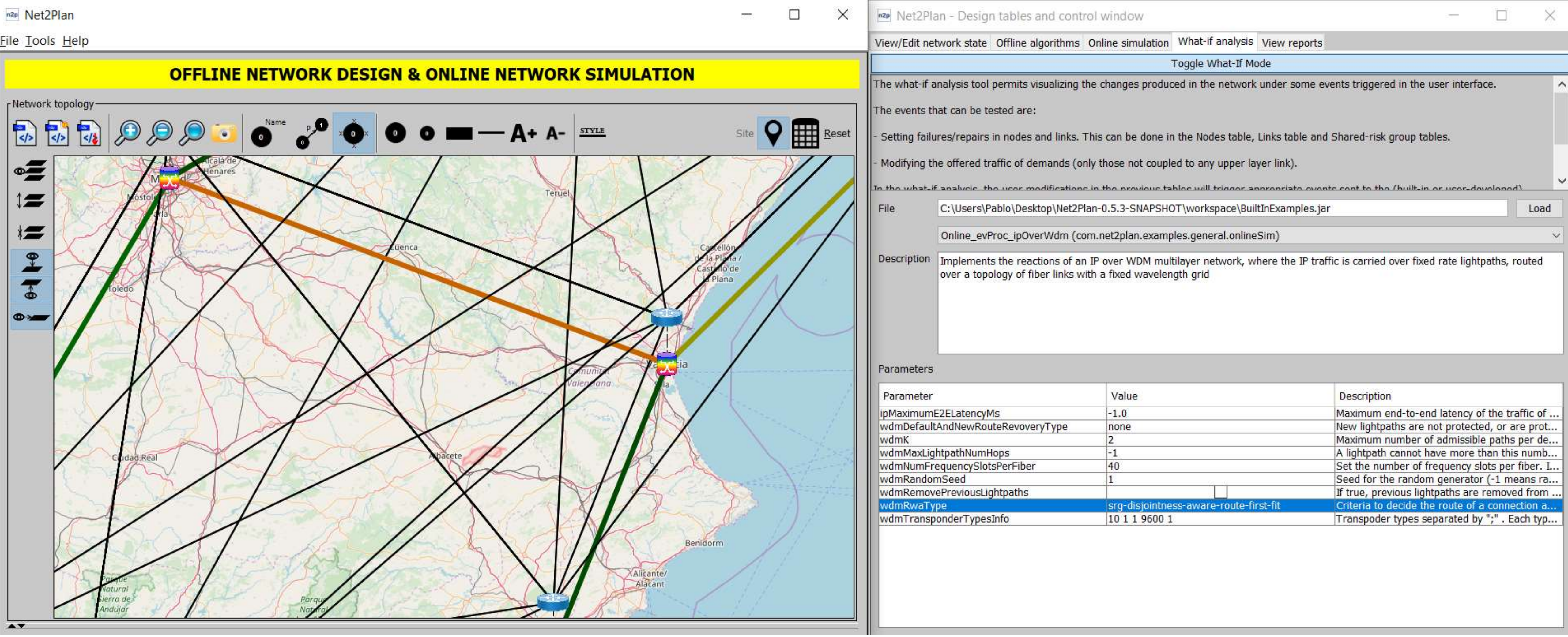
github.com/girtel/net2plan



www.net2plan.com

github.com/girtel/net2plan

- **Multicast traffic**
  - Includes algorithms for solving the *k* minimum cost multicast problem

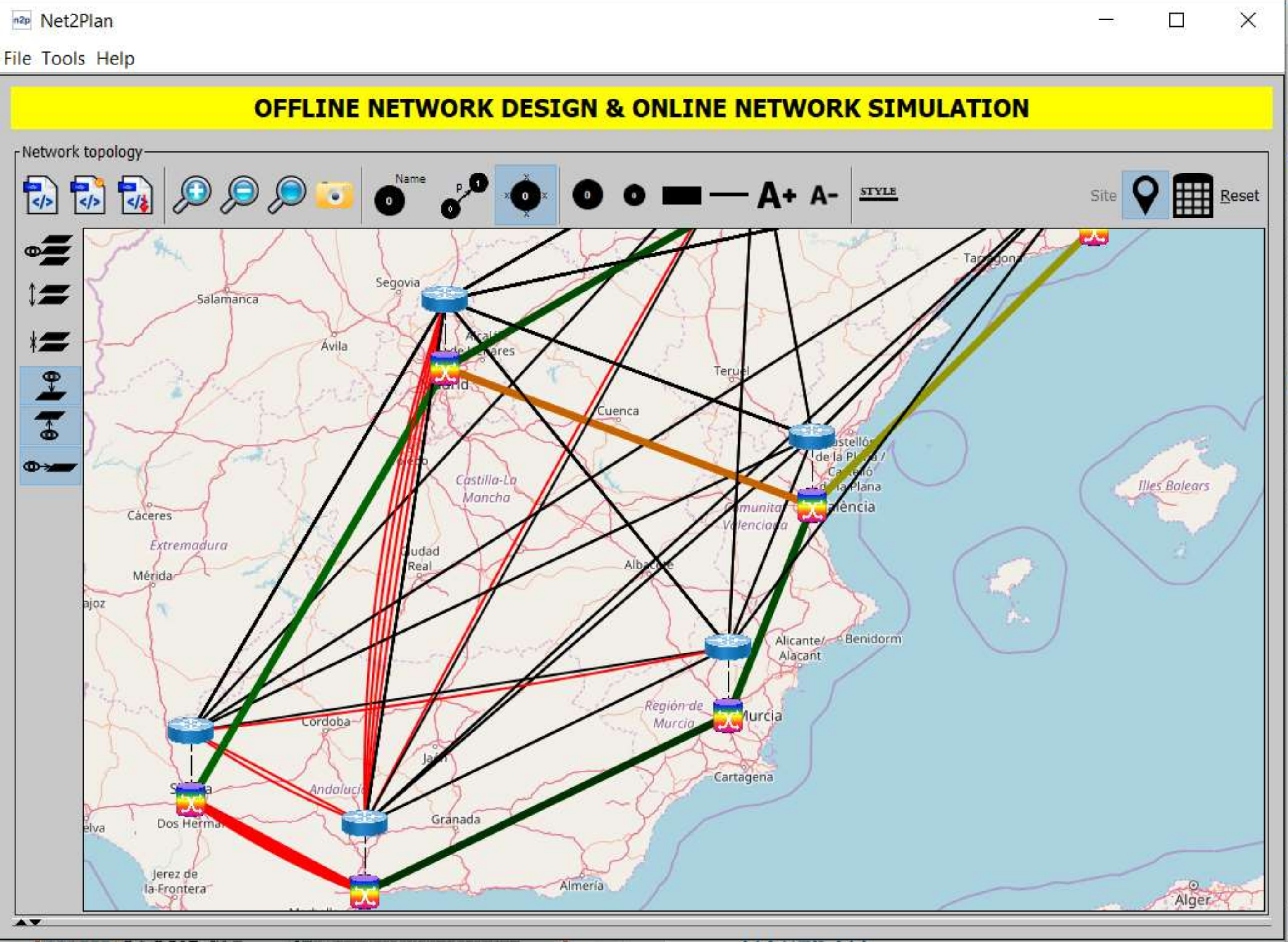


- Multilayer analysis tracking **traffic anomalies** and **failure propagation across layers**
  - Plug in your ***network behavior algorithm*** (or use a built-in one) coding how the network reacts to failures and traffic

net2plan

www.net2plan.com

github.com/girtel/net2plan



Net2Plan - Design tables and control window

View/Edit network state | Offline algorithms | Online simulation | What-if analysis | View reports

Network | Layers | Nodes | Links | Demands | Multicast demands | Routes | Multicast trees | Resources | Shared-risk groups

Number of entries: 16 Reset VFs

Unique id...	Index	Show/Hide	Origin no...	Destinati...	State	Capacity	Carried t...	Occupati...	Utilization	Is bottlen...
9	0	✓	0 (Madrid)	2 (Valencia)	✓	40	3300	33	0.825	✓
10	1	✓	0 (Madrid)	3 (Sevilla)	✓	40	900	9	0.225	
11	2	✓	0 (Madrid)	4 (Zarago...	✓	40	1100	11	0.275	
12	3	✓	1 (Barcelo...	2 (Valencia)	✓	40	2500	25	0.625	
13	4	✓	1 (Barcelo...	4 (Zarago...	✓	40	300	3	0.075	
14	5	✓	2 (Valencia)	0 (Madrid)	✓	40	3300	33	0.825	✓
15	6	✓	2 (Valencia)	1 (Barcelo...	✓	40	2500	25	0.625	
16	7	✓	2 (Valencia)	6 (Murcia)	✓	40	800	8	0.2	
17	8	✓	3 (Sevilla)	0 (Madrid)	✓	40	1400	14	0.35	
18	9	✓	3 (Sevilla)	5 (Málaga)		40	0	0	0	
19	10	✓	4 (Zarago...	0 (Madrid)	✓	40	1000	10	0.25	
20	11	✓	4 (Zarago...	1 (Barcelo...	✓	40	300	3	0.075	
21	12	✓	5 (Málaga)	3 (Sevilla)	✓	40	700	7	0.175	
22	13	✓	5 (Málaga)	6 (Murcia)	✓	40	300	3	0.075	
23	14	✓	6 (Murcia)	2 (Valencia)	✓	40	800	8	0.2	
24	15	✓	6 (Murcia)	5 (Málaga)	✓	40	400	4	0.1	
---	---	---	---	---	---	640.00	19600.00	196.00	---	---

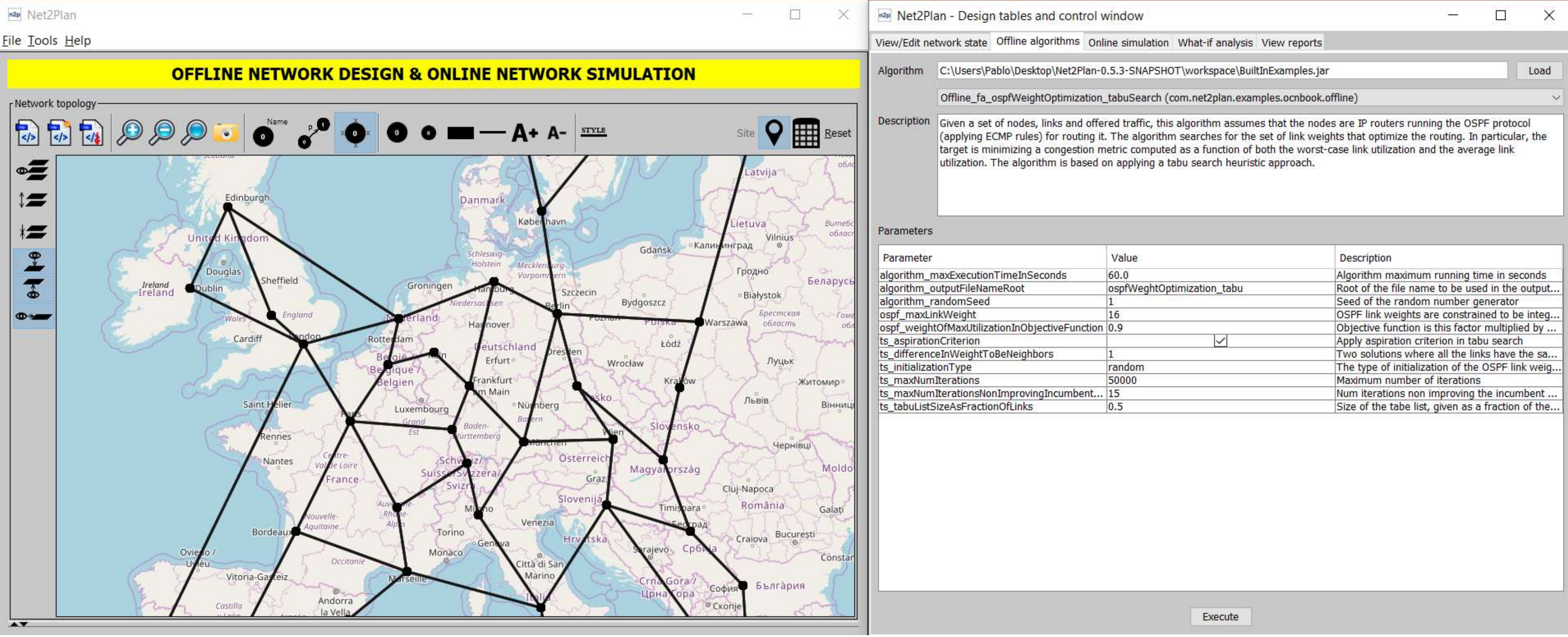
Export tables...

net2plan

www.net2plan.com

github.com/girtel/net2plan

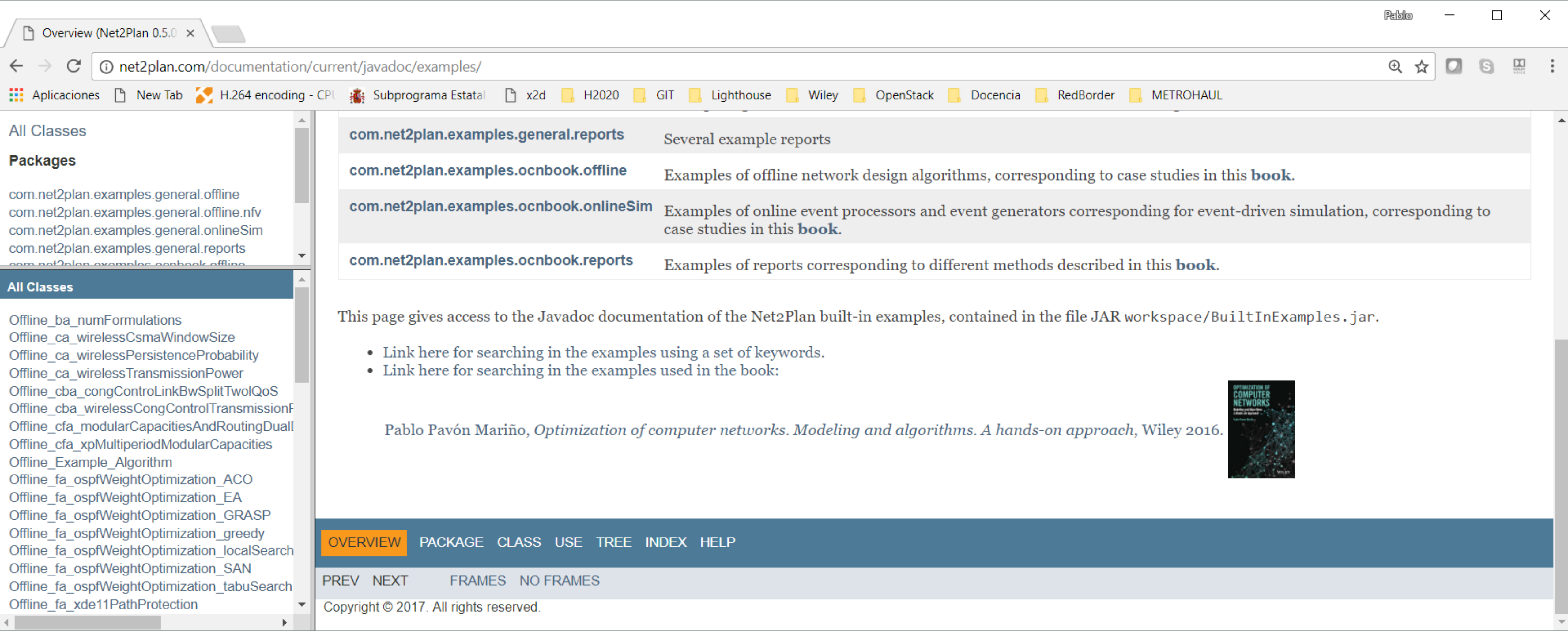
- Multilayer analysis tracking **traffic anomalies** and **failure propagation across layers**
  - Plug in your ***network behavior algorithm*** (or use a built-in one) coding how the network reacts to failures and traffic
  - **WHAT-IF**: Then, manually create failures/repairs and/or traffic shifts, and see the network reaction in all the layers



- **Offline algorithms (e.g. for capacity planning)**
  - Plug in your algorithm (or use a built-in one) that makes network redesigns
  - Check the repository of algorithms in the Javadoc!!

www.net2plan.com

github.com/girtel/net2plan

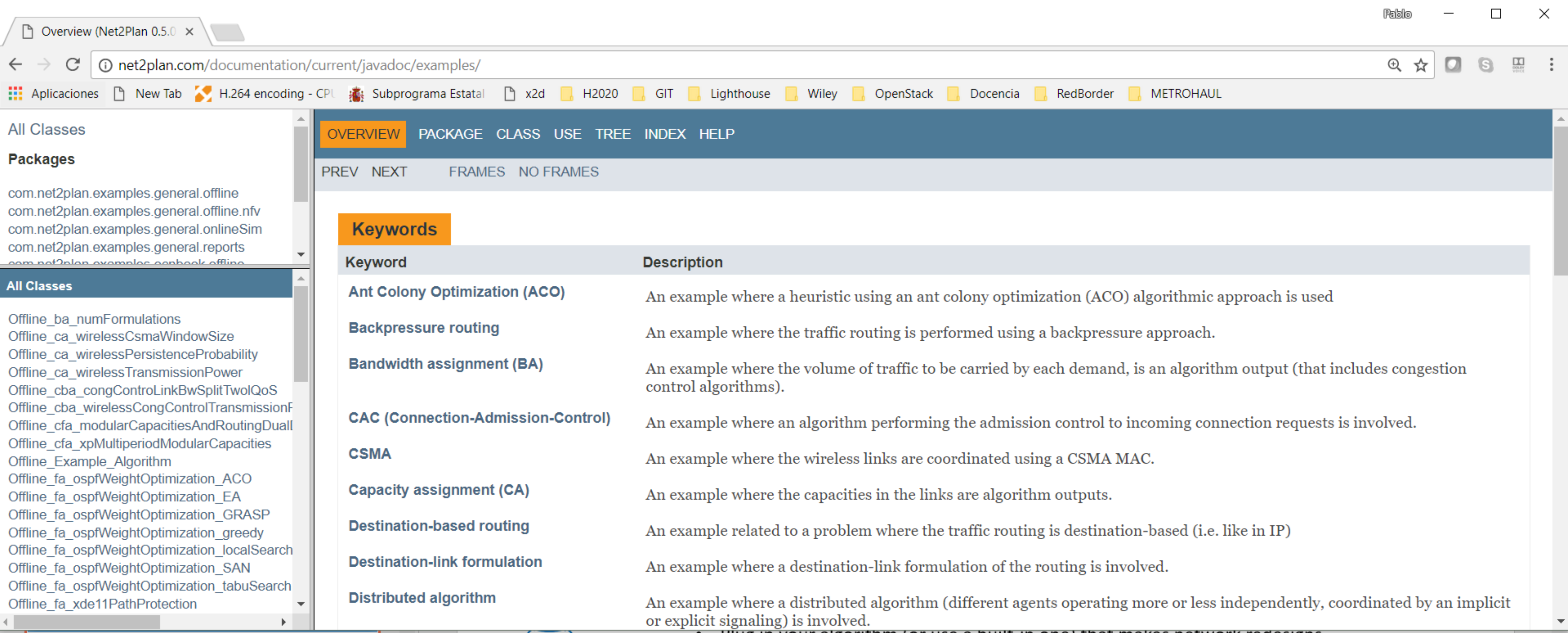


net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Offline algorithms (e.g. for capacity planning)**
  - Plug in your algorithm (or use a built-in one) that makes network redesigns
  - Check the repository of algorithms in the Javadoc!!

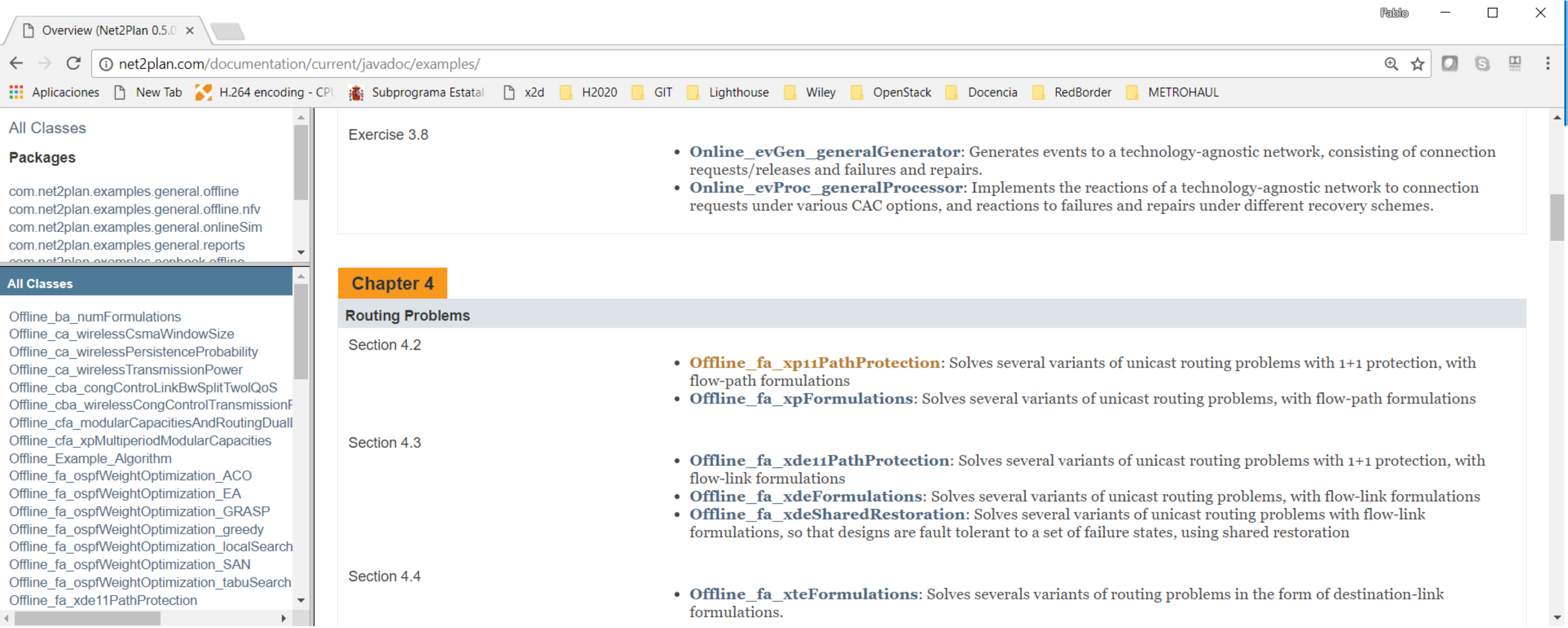


net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Offline algorithms (e.g. for capacity planning)**
  - Plug in your algorithm (or use a built-in one) that makes network redesigns
  - Check the repository of algorithms in the Javadoc!!
    - Includes indexing by keyword to search for algorithms

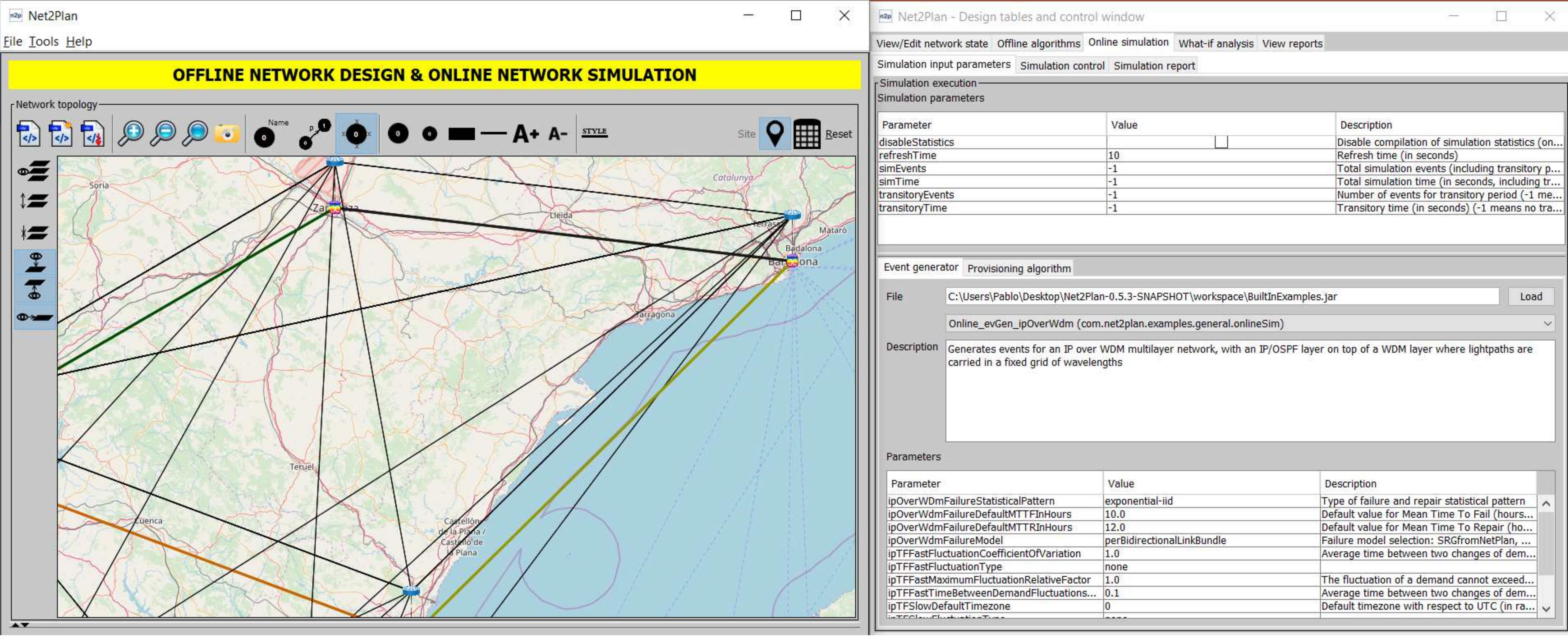


net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Offline algorithms (e.g. for capacity planning)**
  - Plug in your algorithm (or use a built-in one) that makes network redesigns
  - Check the repository of algorithms in the Javadoc!!
    - Includes indexing by keyword to search for algorithms
    - For book readers: includes links to sections describing the maths behind

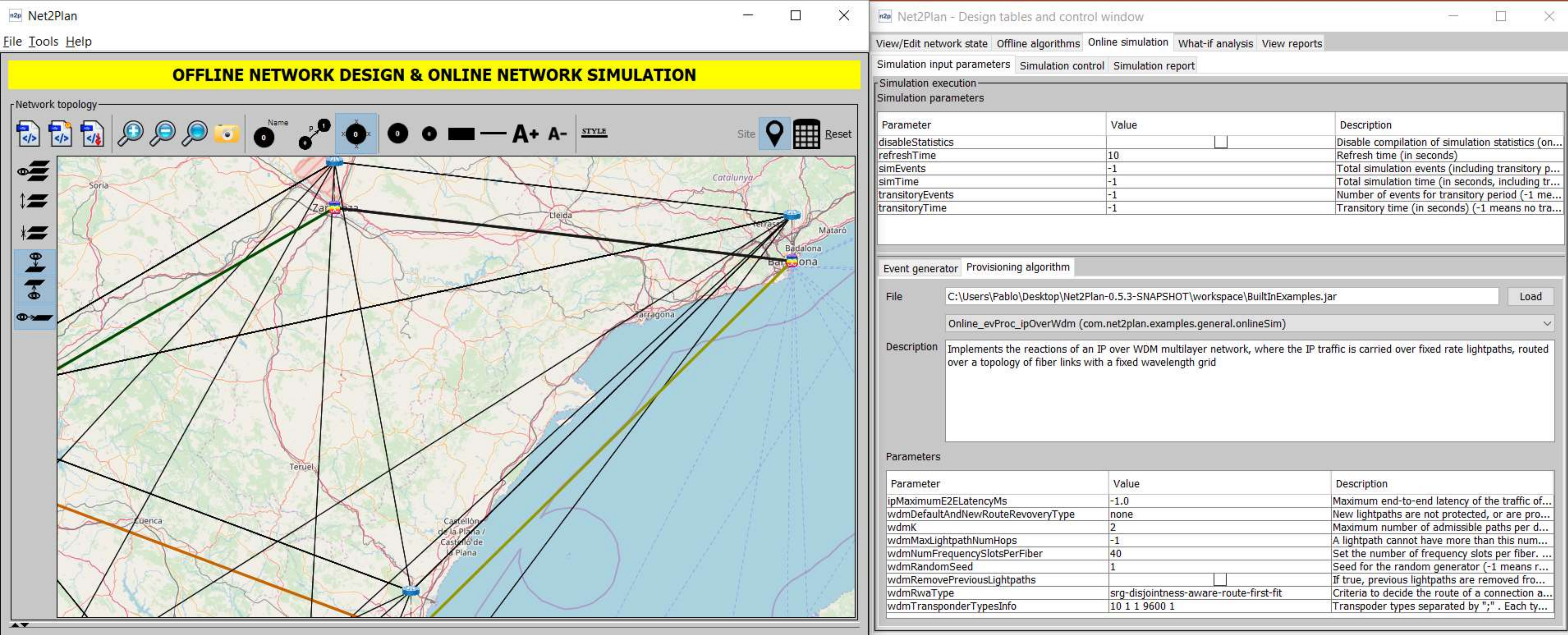


net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Online (provisioning) algorithms for event-driven network simulation**
  - Plug in your event generation algorithm (or use a built-in one) that produces traffic shifts, failures and repairs etc.

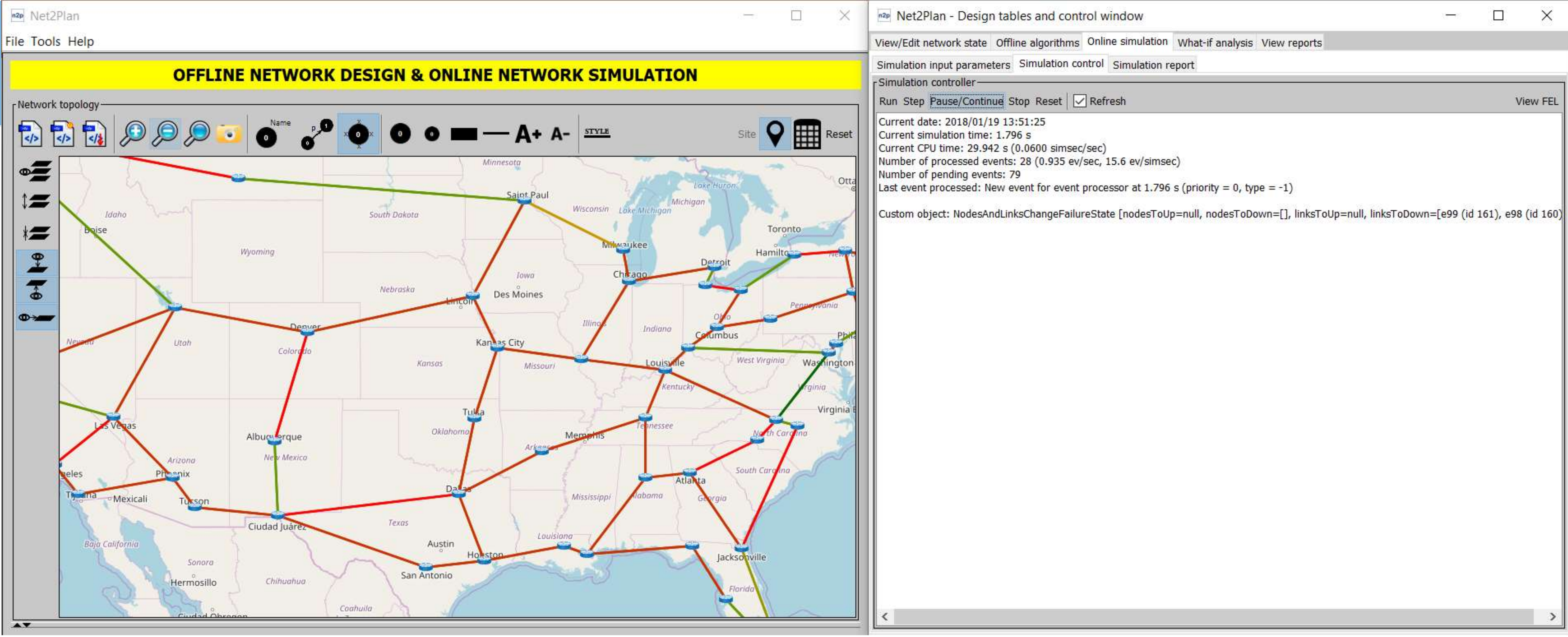


net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Online (provisioning) algorithms for event-driven network simulation**
  - Plug in your event generation algorithm (or use a built-in one) that produces traffic shifts, failures and rapairs etc.
  - Plug in your ***network behavior algorithm*** coding network reaction (same algorithm as in *what-if*, no need to code a new one!!!)

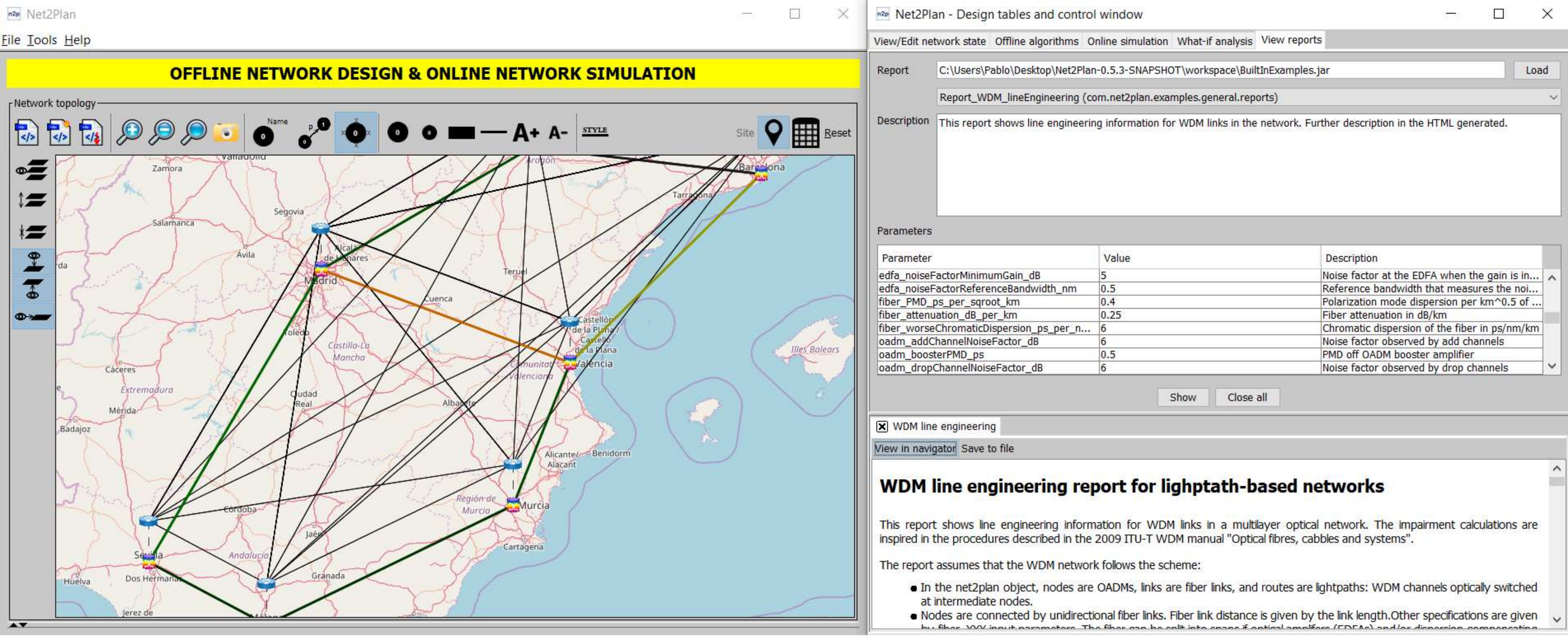


- **Online (provisioning) algorithms for event-driven network simulation**
  - Run, step, continue, pause, stop the simulation
  - See the simulation report:
    - Tech-agnostic statistic are automatically computed by Net2Plan
    - User algorithms can create custom (e.g. tech-specific) reports

net2plan

www.net2plan.com

github.com/girtel/net2plan



net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Reporting**. Plug in your report algorithm, or use a built-in one. Some interesting built-in reports:
  - WDM line engineering, OSNR (using GN model), PMD, CD, power at link & lightpath level
  - RSA inspector: Spectrum occupation in the fibers
  - Availability analysis: Estimates service level availability (includes error margin)

WDM line engineering in ×

file:///C:/Users/Pablo/AppData/Local/Temp/tmp_5054036585432895723.html

Aplicaciones New Tab H.264 encoding - CPU Subprograma Estatal x2d H2020 GIT Lighthouse Wiley OpenStack Docencia RedBorder METROHAUL

PER LINK INFORMATION SUMMARY - Signal metrics at the input of end OADM

Link #	Length (km)	# EDFAs	# DCMs	Chromatic Dispersion (ps/nm)	OSNR (dB)	Power per WDM channel (dBm)	Polarization Mode Dispersion (ps)	Warnings
e0 (id 9) (Madrid --> Valencia)	301.92	3	3	0	27.64	-19	7.14	
e1 (id 10) (Madrid --> Sevilla)	391.43	4	3	0	25.89	-19	8.1	
e2 (id 11) (Madrid --> Zaragoza)	272.44	3	2	-0	30.95	-18.11	6.77	
e3 (id 12) (Barcelona --> Valencia)	303.36	3	3	0	27.52	-19	7.16	
e4 (id 13) (Barcelona --> Zaragoza)	256.51	3	2	0	30.95	-14.13	6.58	
e5 (id 14) (Valencia --> Madrid)	301.92	3	3	0	27.64	-19	7.14	
e6 (id 15) (Valencia --> Barcelona)	303.36	3	3	0	27.52	-19	7.16	
e7 (id 16) (Valencia --> Murcia)	177.19	2	2	0	32.68	-11.3	5.51	
e8 (id 17) (Sevilla --> Madrid)	391.43	4	3	0	25.89	-19	8.1	
e9 (id 18) (Sevilla --> Málaga)	157.56	1	1	118.36	34.76	-19	5.14	
e10 (id 19) (Zaragoza --> Madrid)	272.44	3	2	-0	30.95	-18.11	6.77	
e11 (id 20) (Zaragoza --> Barcelona)	256.51	3	2	0	30.95	-14.13	6.58	
e12 (id 21) (Málaga --> Sevilla)	157.56	1	1	118.36	34.76	-19	5.14	
e13 (id 22) (Málaga --> Murcia)	322.94	3	3	0	25.9	-19	7.38	
e14 (id 23) (Murcia --> Valencia)	177.19	2	2	0	32.68	-11.3	5.51	
e15 (id 24) (Murcia --> Málaga)	322.94	3	3	0	25.9	-19	7.38	

PER ROUTE INFORMATION SUMMARY - Signal metrics at the transponder

Route #	Length (km)	# EDFAs	# DCMs	Chromatic Dispersion (ps/nm)	OSNR (dB)	Power per WDM channel (dBm)	Polarization Mode Dispersion (ps)	Warnings
r0 (id 155) (Madrid --> Barcelona)	605.29	6	6	0	19.96	-19	10.11	

net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Reporting**. Plug in your report algorithm, or use a built-in one. Some interesting built-in reports:
  - **WDM line engineering**: OSNR (using GN model), PMD, CD, power at link & lightpath level
  - RSA inspector: Spectrum occupation in the fibers
  - Availability analysis: Estimates service level availability (includes error margin)

TIP – Open Optical Packet Transport

TELECOM INFRA PROJECT

http://telecominfraproject.com/

- Photonic Simulation Environment (PSE) Group within Open Optical Packet Transport (OOPT) working on OpenSource Optical Link Emulator:
  - GN model for Non Linear impairment estimation
  - Python code delivery: https://github.com/Telecominfraproject/gnpy/
  - Multi-vendor approach

net2plan

- Net2Plan incorporates two models for estimating nonlinear impairments, one of them applying a flavor of the Gaussian Noise model (thanks POLITO GN-group, thanks OOPT-PSE group!!)
- **Checkout OOPT-PSE GIT repository for a Python-based library for optical quality of transmission estimations!! Great work there!!**

WDM line engineering in × | WDM Lightpath Routing ×

file:///C:/Users/Pablo/AppData/Local/Temp/tmp_2605642235536216179.html

Aplicaciones | New Tab | H.264 encoding - CPU | Subprograma Estatal | x2d | H2020 | GIT | Lighthouse | Wiley | OpenStack | Docencia | RedBorder | METROHAUL

% of carried traffic with at least one backup path	0 %

PER FIBER INFORMATION SUMMARY

This table shows information for each fiber. In particular, the slots occupied, with a link to the lightpaths occupying it, either for regular lightpaths (L), or lightpaths defined as protection segments (P) that reserve slots:

- Black: The slot number is higher than the capacity declared for the link, and is not assigned to any lightpath.
- White: The slot is within the fiber capacity, and is not assigned to any lightpath.
- Green: The slot is within the fiber capacity, and is occupied by one regular lightpath and assigned to no backup lightpath.
- Yellow: The slot is within the fiber capacity, and is occupied by zero regular lightpaths and assigned to one backup lightpath.
- Red: The slot is within the fiber capacity, and is occupied by more than one lightpath (summing regular and backup), or is outside the link capacity and is assigned to at leastone lightpath.

Fiber #	Origin node	Dest. node	% slots used	Ok?	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0 (id: 9)	n0 (Madrid)	n2 (Valencia)	0.825	Yes	L0	L2	L4	L6	L8	L10	L12	L14	L16	L18	L20	L22	L24	L26	L28	L32	L36	L40
1 (id: 10)	n0 (Madrid)	n3 (Sevilla)	0.35	Yes	L34	L42	L50	L52	L60	L62	L74	L78	L92	L102	L119	L122						
2 (id: 11)	n0 (Madrid)	n4 (Zaragoza)	0.275	Yes	L30	L38	L46	L56	L68	L84	L116		L91			L118	L123					
3 (id: 12)	n1 (Barcelona)	n2 (Valencia)	0.625	Yes	L1	L3	L5	L7	L9	L11	L13	L15	L17	L19	L21	L70	L25	L86	L29	L98	L37	L100
4 (id: 13)	n1 (Barcelona)	n4 (Zaragoza)	0.075	Yes	L76	L96																
5 (id: 14)	n2 (Valencia)	n0 (Madrid)	0.825	Yes	L1	L3	L5	L7	L9	L11	L13	L15	L17	L19	L21	L23	L25	L27	L29	L33	L37	L41
6 (id: 15)	n2 (Valencia)	n1 (Barcelona)	0.625	Yes	L0	L2	L4	L6	L8	L10	L12	L14	L16	L18	L20	L71	L24	L87	L28	L99	L36	L101
7 (id: [illegible]	n2 [illegible]	n6 (Murcia)	0.2	Yes	L106	L112														L98		L100

Fiber #	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0 (id: 9)	L44	L48	L54	L58	L64	L66	L72	L80	L82	L88	L95	L108	L111	L114	L120				
1 (id: 10)											L90			L110					
2 (id: 11)												L94							
3 (id: 12)	L45	L128	L55			L67			L83		L90				L115				
4 (id: 13)		L121																	
5 (id: 14)	L45	L49	L55	L59	L65	L67	L73	L81	L83	L89	L90	L94	L109	L110	L115				
6 (id: 15)	L44	L121	L54	L129		L66			L82					L114					
7 (id: [illegible]					L64			L80				L108			L120				

net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Reporting**. Plug in your report algorithm, or use a built-in one. Some interesting built-in reports:
  - WDM line engineering: OSNR (using GN model), PMD, CD, power at link & lightpath level
  - **RSA inspector**: Spectrum occupation in the fibers
  - Availability analysis: Estimates service level availability (includes error margin)

WDM line engineering in × | WDM Lightpath Routing × | Single-SRG failure analys ×

file:///C:/Users/Pablo/AppData/Local/Temp/tmp_7920470694323556656.html

Aplicaciones | New Tab | H.264 encoding - CPU | Subprograma Estatal | x2d | H2020 | GIT | Lighthouse | Wiley | OpenStack | Docencia | RedBorder | METROHAUL

Layer WDM, index = 0, id = 1

Unicast traffic

SRG Index failed	Offered traffic	Blocked traffic (%)	Offered traffic traversing oversubscribed links (%)	Offered traffic of demands with excessive latency (%)	Total blocked traffic [out of contract] (%)	% of demands fully ok
No failure	13000.000	0.000 (0.000 %)	0.000 (0.000 %)	0.000 (0.000 %)	0.000 (0.000 %)	(100.000 %)
0	13000.000	6600.000 (50.769 %)	0.000 (0.000 %)	0.000 (0.000 %)	6600.000 (50.769 %)	(49.231 %)
1	13000.000	2800.000 (21.538 %)	0.000 (0.000 %)	0.000 (0.000 %)	2800.000 (21.538 %)	(78.462 %)
2	13000.000	2200.000 (16.923 %)	0.000 (0.000 %)	0.000 (0.000 %)	2200.000 (16.923 %)	(83.077 %)
3	13000.000	5000.000 (38.462 %)	0.000 (0.000 %)	0.000 (0.000 %)	5000.000 (38.462 %)	(61.538 %)
4	13000.000	600.000 (4.615 %)	0.000 (0.000 %)	0.000 (0.000 %)	600.000 (4.615 %)	(95.385 %)
5	13000.000	1600.000 (12.308 %)	0.000 (0.000 %)	0.000 (0.000 %)	1600.000 (12.308 %)	(87.692 %)
6	13000.000	1400.000 (10.769 %)	0.000 (0.000 %)	0.000 (0.000 %)	1400.000 (10.769 %)	(89.231 %)
7	13000.000	800.000 (6.154 %)	0.000 (0.000 %)	0.000 (0.000 %)	800.000 (6.154 %)	(93.846 %)

Layer IP, index = 1, id = 285

Unicast traffic

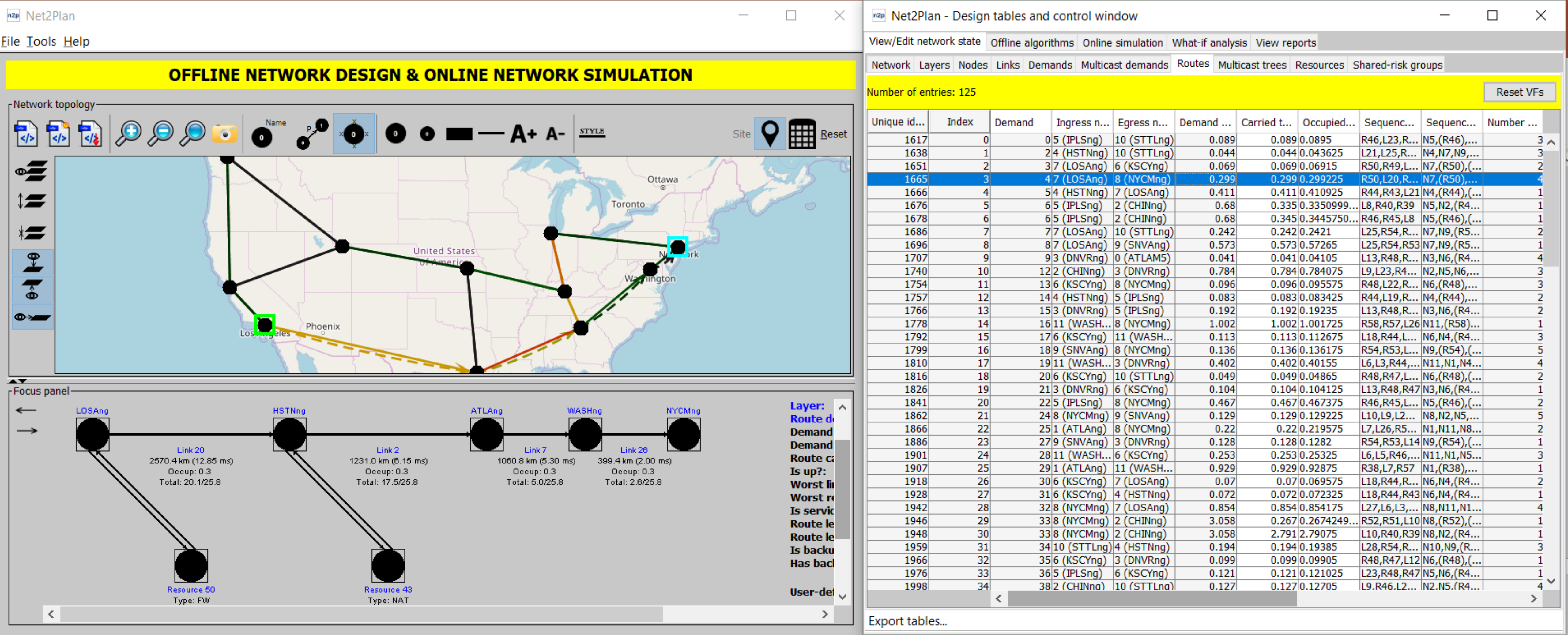
SRG Index failed	Offered traffic	Blocked traffic (%)	Offered traffic traversing oversubscribed links (%)	Offered traffic of demands with excessive latency (%)	Total blocked traffic [out of contract] (%)	% of demands fully ok
No failure	10000.000	0.000 (0.000 %)	0.000 (0.000 %)	0.000 (0.000 %)	0.000 (0.000 %)	(100.000 %)
0	10000.000	0.000 (0.000 %)	9320.197 (93.202 %)	0.000 (0.000 %)	9320.197 (93.202 %)	(33.333 %)
1	10000.000	0.000 (0.000 %)	7727.512 (77.275 %)	0.000 (0.000 %)	7727.512 (77.275 %)	(40.476 %)
2	10000.000	0.000 (0.000 %)	5605.399 (56.054 %)	0.000 (0.000 %)	5605.399 (56.054 %)	(61.905 %)
3	10000.000	0.000 (0.000 %)	6503.431 (65.034 %)	0.000 (0.000 %)	6503.431 (65.034 %)	(54.762 %)

net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Reporting**. Plug in your report algorithm, or use a built-in one. Some interesting built-in reports:
  - WDM line engineering: OSNR (using GN model), PMD, CD, power at link & lightpath level
  - RSA inspector: Spectrum occupation in the fibers
  - **Availability analysis**: Estimates service level availability (includes error margin)

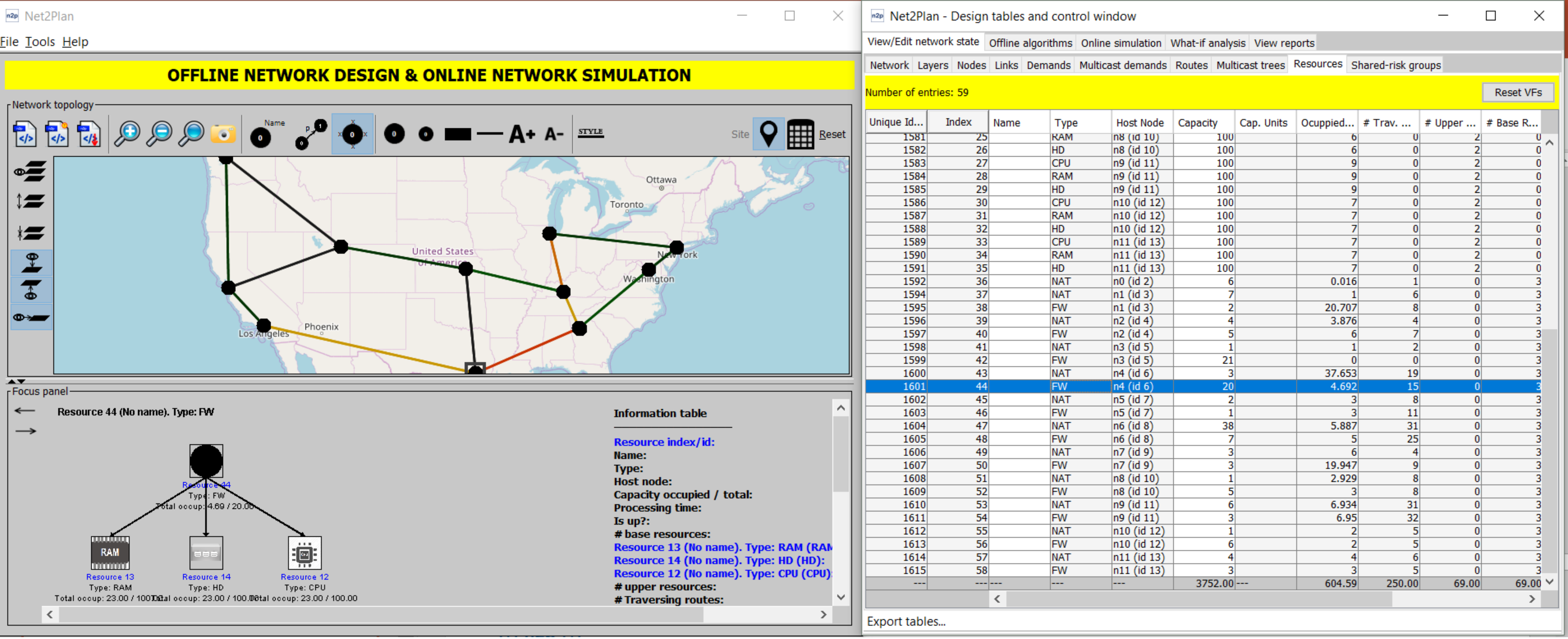


net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Support for offline/online algorithms allocating IT & Network resources in NFV context**
  - Concept of Resource (e.g. vFirewall) in the nodes, traversable by flows, with a given capacity (e.g. Gbps of traversing flows), consumes other resources (e.g. 1 CPU, 8 GB of RAM, 10 GB HD)
  - Service chain is a route from A to B, that needs to traverse resources of a given type in a particular order (e.g. first vNAT, then vFirewall, then vMonitor)

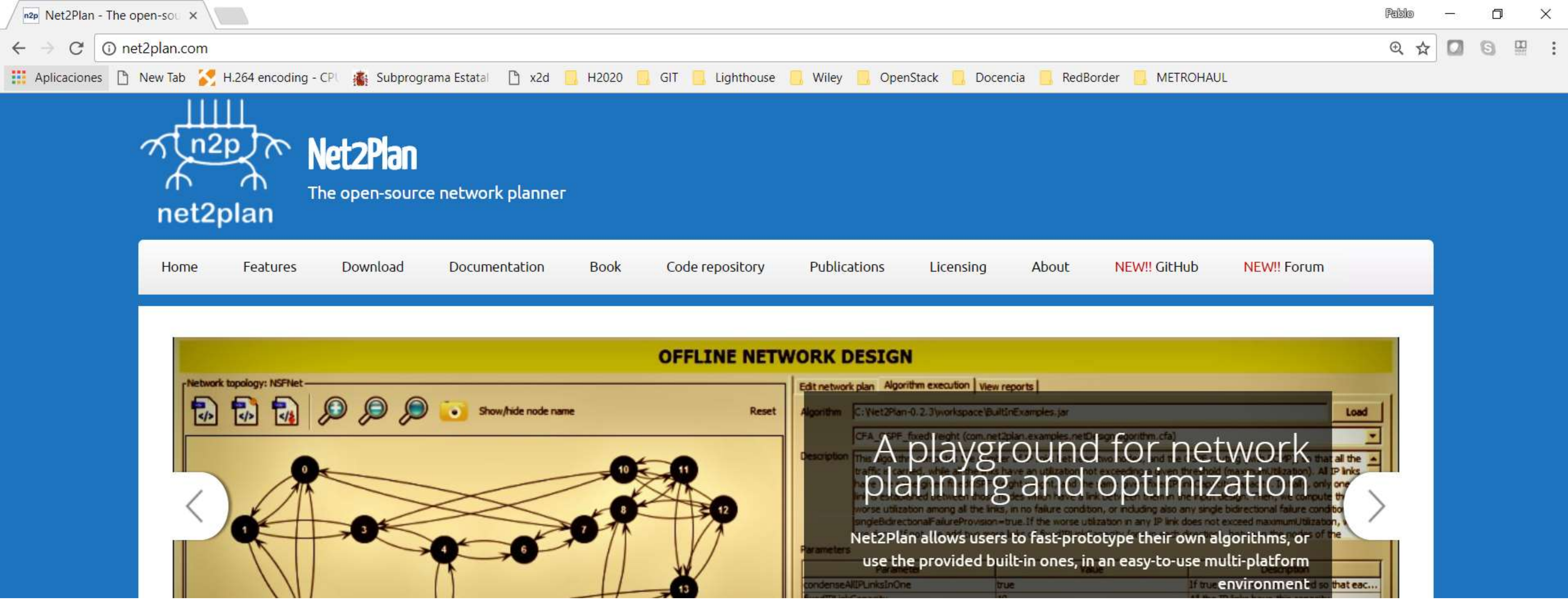


net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Support for offline/online algorithms allocating IT & Network resources in NFV context**
  - Includes built-in algorithms in this scope
  - Includes a *utils* library with algorithms for solving the *k* minimum cost service chain problem

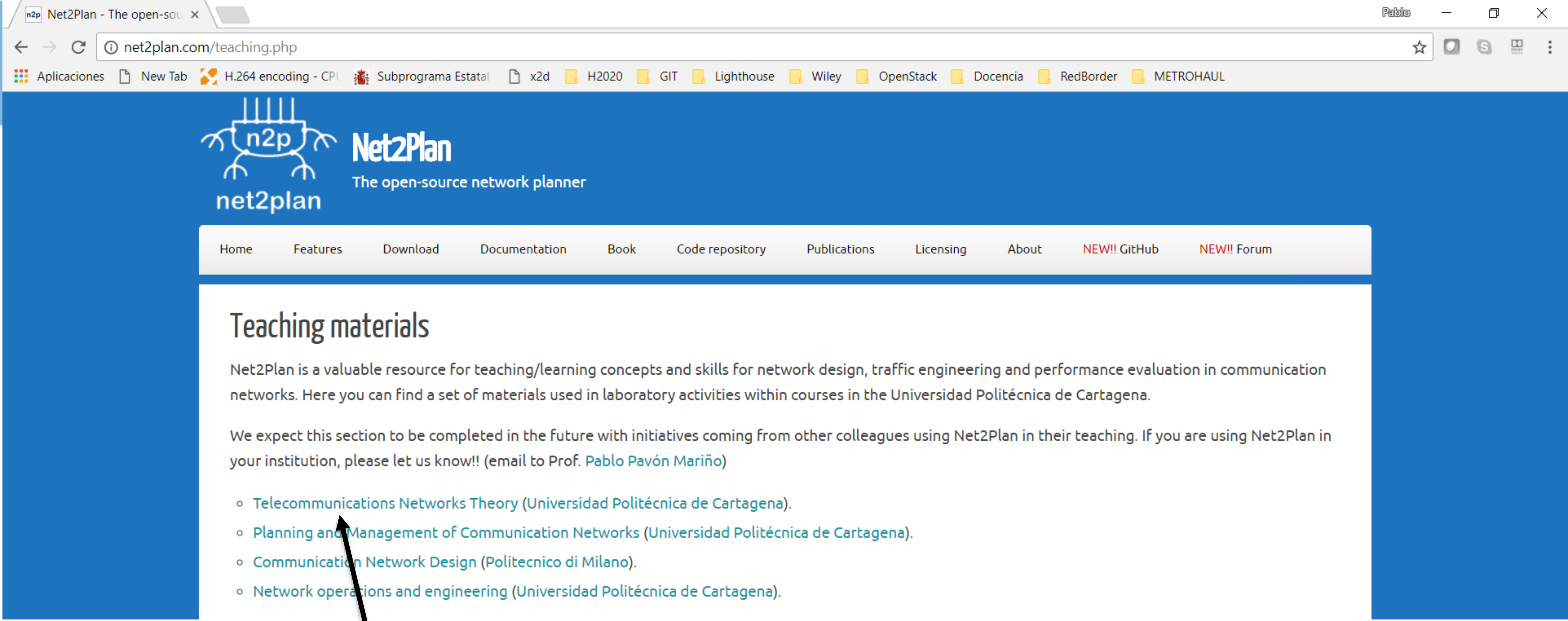


net2plan

www.net2plan.com

github.com/girtel/net2plan

- **How to obtain more information for...**
  - **Using it:** Users guide
  - **Developing algorithms:** Javadoc
  - **The Math:** the book is suited for that



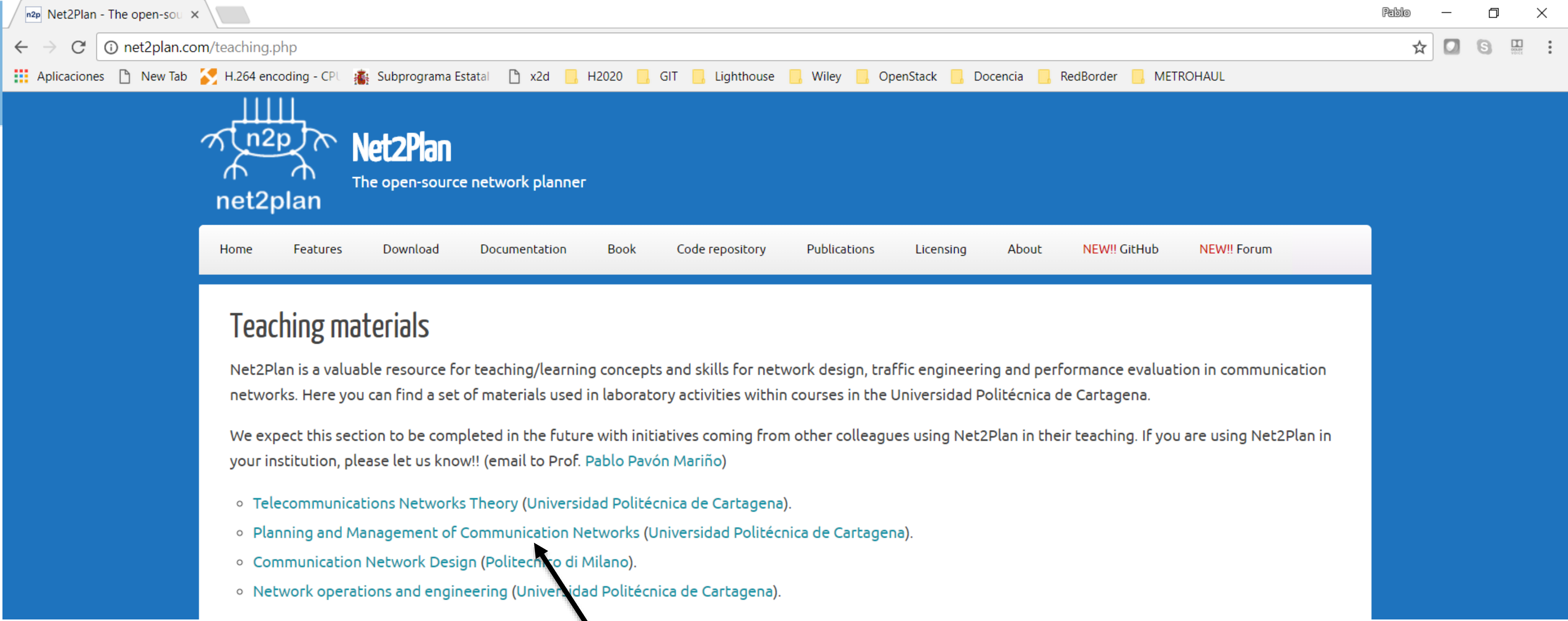
net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Teaching materials**

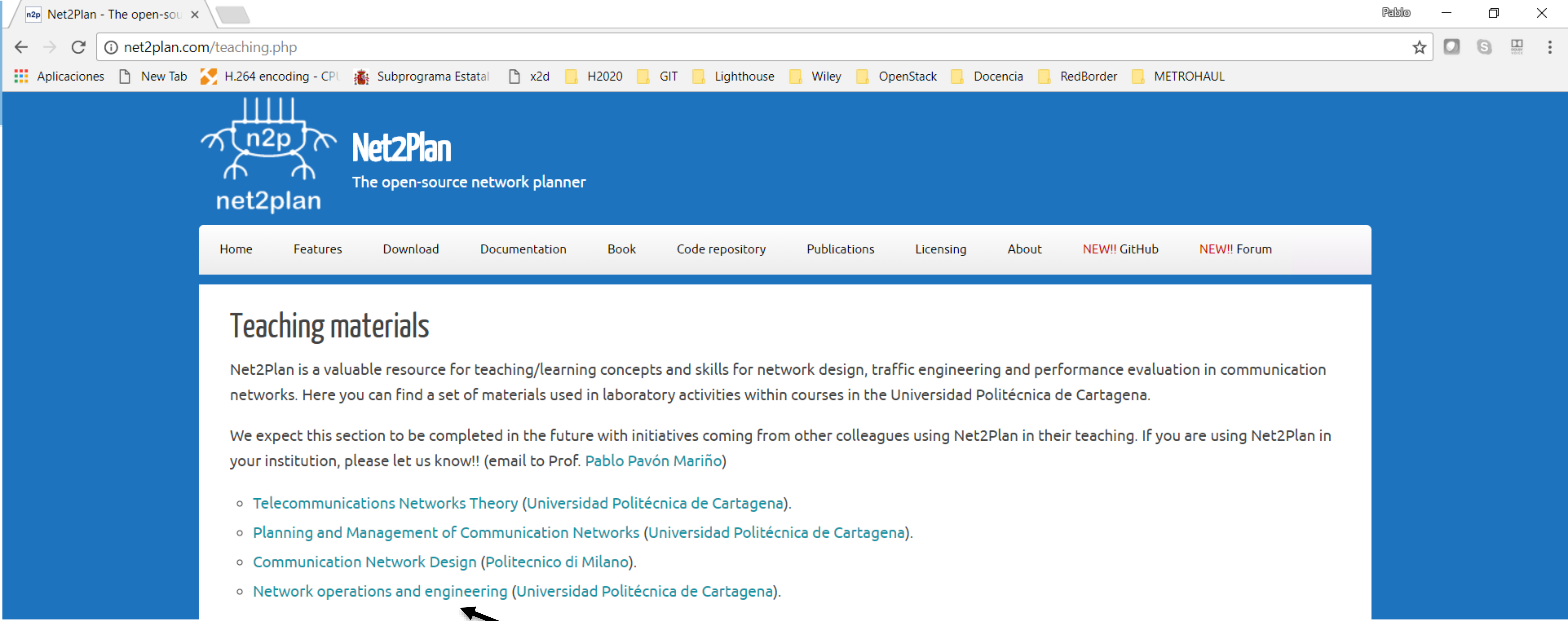
  Strongly recomended for starters!! Includes 10 lab work sessions + solutions.
  Content: installing + basic use + developing algorithms based on formulations
  Level: 2nd year BSc students in Telecommunication degrees



- **Teaching materials**

  Focused on heuristics (not formulation based algorithms)

n2p
net2plan
www.net2plan.com
github.com/girtel/net2plan



net2plan

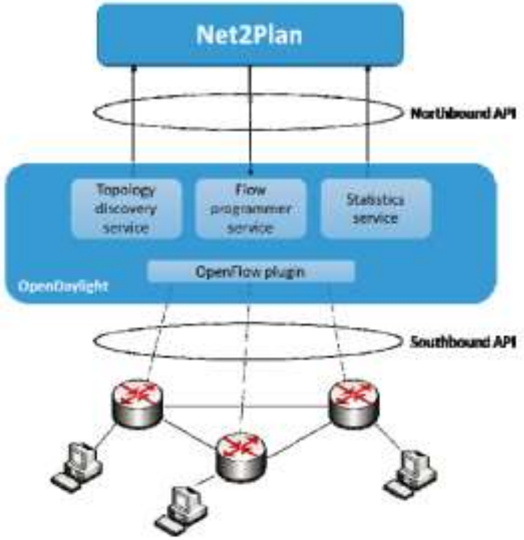
www.net2plan.com

github.com/girtel/net2plan

- **Teaching materials**

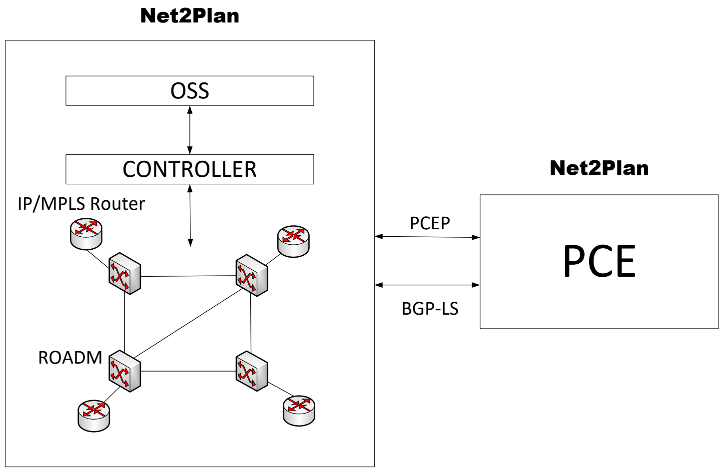
  Long case studies planning IP over WDM using Net2Plan using real equipment catalogues. EDFA placement, IP chassis & cards provisioning, virtual topology design, service level availability, total cost evaluations

2014 - ODL



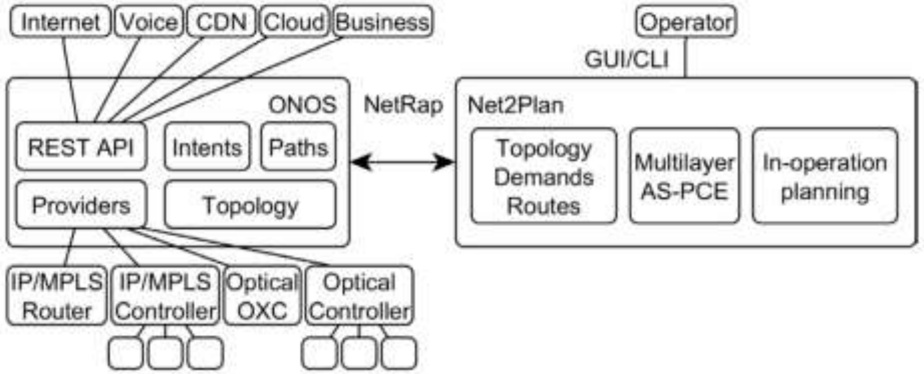
J.-L. Izquierdo-Zaragoza, A. Fernandez-Gambin, J.-J. Pedreno-Manresa and P. Pavon-Marino, “Leveraging Net2Plan planning tool for network orchestration in OpenDaylight”, in *SaCoNeT 2014*

2016 – ABNO



J.L. Izquierdo-Zaragoza, J.J. Pedreno-Manresa, P. Pavon-Marino, O. Gonzalez-de-Dios and V. Lopez, "Dynamic Operation of an IP/MPLS-over-WDM Network Using an Active Stateful BGP/LS-Enabled Multilayer PCE", in *ICTON 2016*

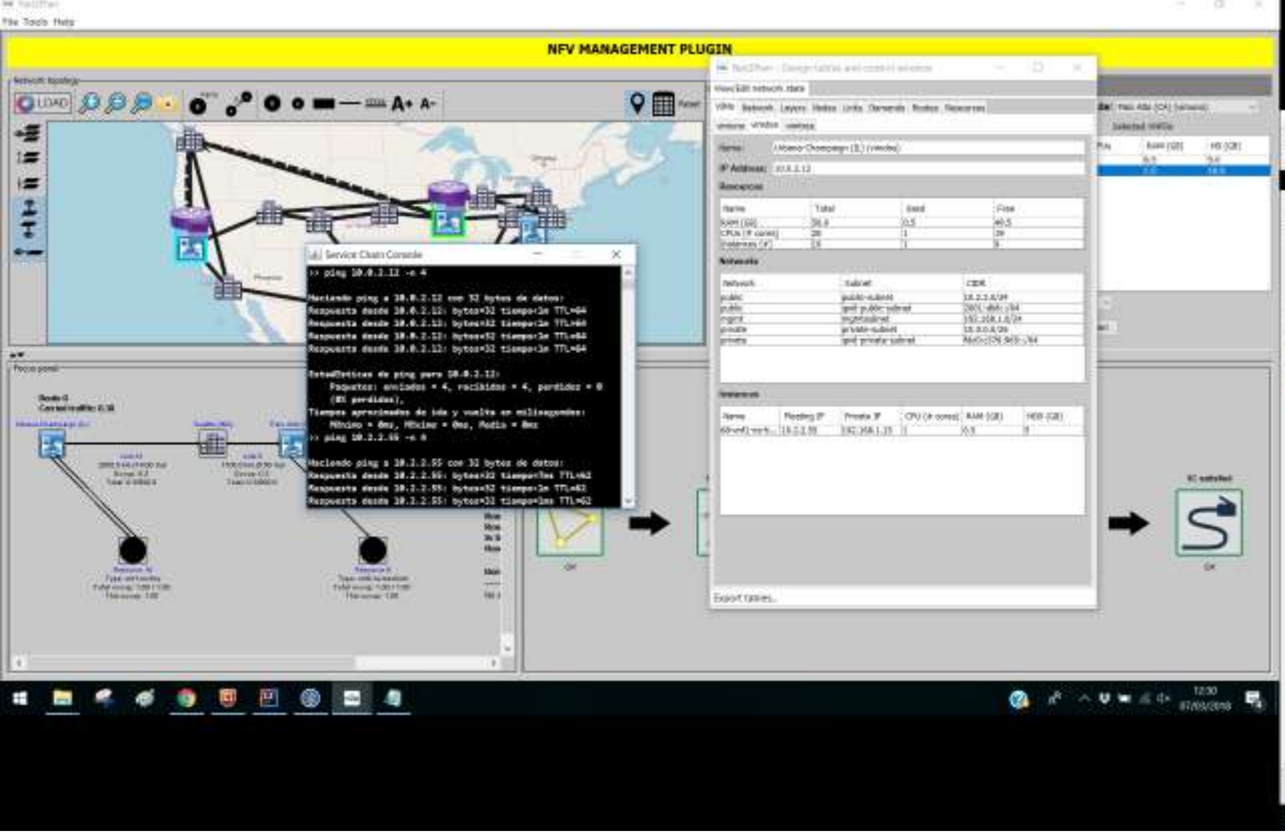
2017 - ONOS

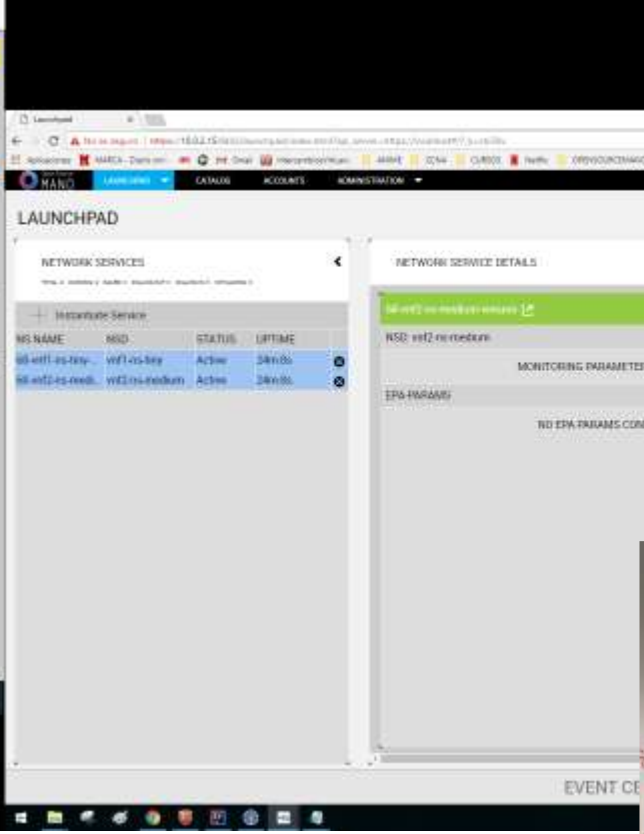


Pontus Sköldström, Ćiril Rožić and Jose-Juan Pedreno-Manresa, “Making powerful friends: Introducing ONOS and Net2Plan to each other,” in 19th International Conference on Transparent Optical Networks (ICTON 2017).

net2plan
www.net2plan.com
github.com/girtel/net2plan

- **Net2Plan and SDN**
  - Several prototypes and PoCs have been made





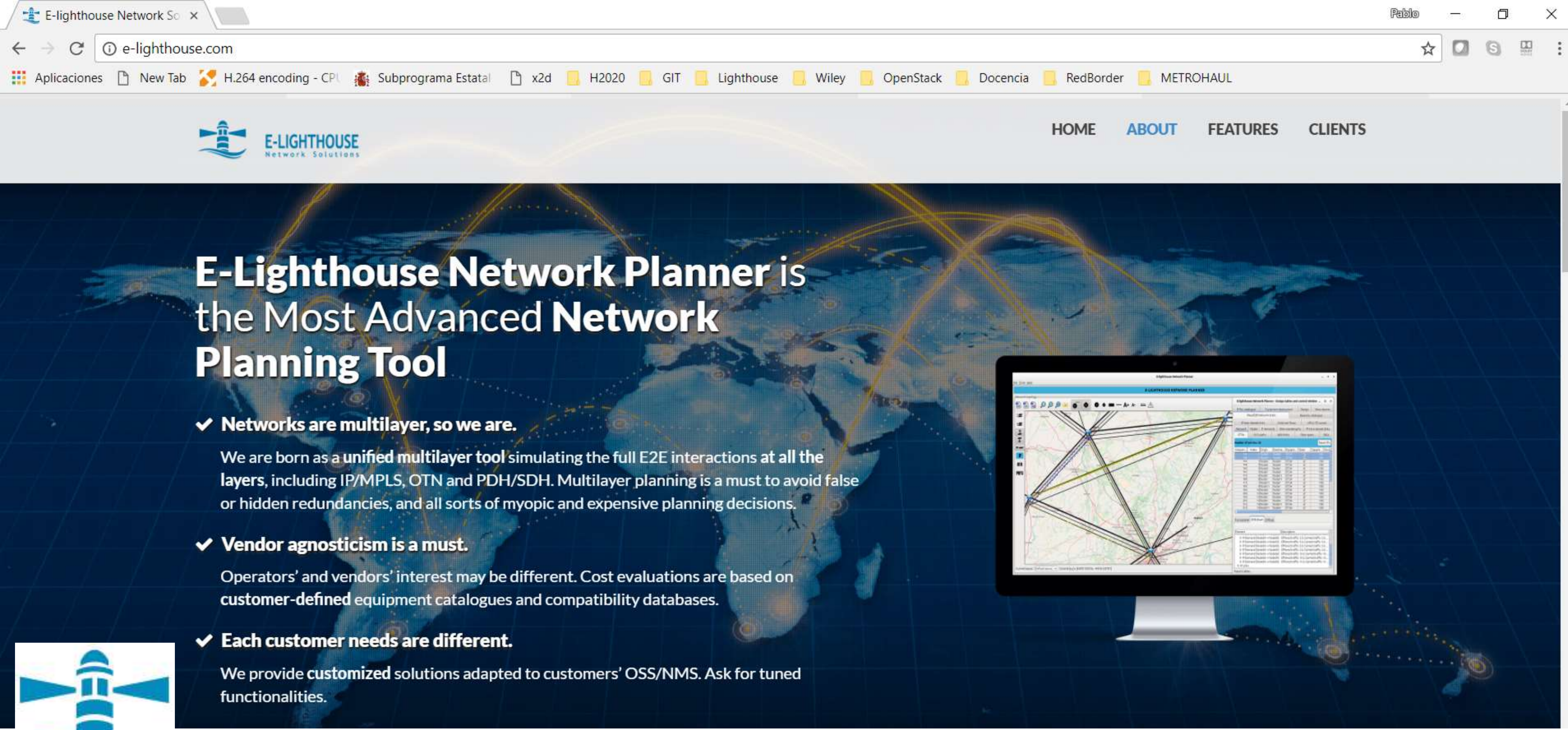
F.J. Moreno-Muro, C. San-Nicolas-Martinez, E. Martin-Seoane, M. Garrich, P. Pavon-Marino, O. Gonzalez de Dios, V. López, "Joint Optimal Service Chain Allocation, VNF instantiation and Metro Network Resource Management Demonstration", OFC 2018 (SDN/NFV Demo Zone)

net2plan

www.net2plan.com

github.com/girtel/net2plan

- **Net2Plan and SDN/NFV**
  - **OFC 2018 SDN/NFV Demo zone**. Connection of Net2Plan to **OpenStack** & **OSM** for making optimized allocations of service chains in multi-VIM environment
  - **Demonstrating joint allocation of IT (CPU, RAM, HD) and network resources**



- **2017. Founding of E-lighthouse Network Solutions**
  - E-lighthouse Network Planner leverages in Net2Plan experience

1. Introduction
2. Use cases
3. Theoretical limits, heuristics, solvers
4. Network optimization software. Net2Plan
5. **Wrap up**

Conclusions

- SDN/NFV means an unprecedented network control, for an unprecedented resource dynamicity

The era of network optimization

- **Stay focused!!** Many use cases appear now in PRODUCTION networks, where optimization is decisive and manual provisioning is unmanageable, let's go for them!
- There are some good open-source resources to exploit, let's use them!

Thanks!!!

Pablo Pavón Mariño (pablo.pavon@upct.es, ppavon@e-lighthouse.com)
Miquel Garrich (miquel.garrich@upct.es)
Francisco Javier Moreno Muro (javier.moreno@upct.es)

Universidad Politécnica de Cartagena

E-LIGHTHOUSE Network Solutions