



Universidad
Politécnica
de Cartagena



OPERACIÓN E INGENIERÍA DE RED

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN
CURSO 2019-2020

Lab work #2. IP monitoring, forecasting and traffic matrix estimation with Net2Plan NIW framework

(1 session)

Author:

Pablo Pavón Mariño

1 Objectives

The goals of this lab work are:

1. Understand and gain key network planning capabilities related to the processing of IP traffic monitoring traces, to produce traffic demand forecasts, that may be used for e.g. capacity planning. Use Net2Plan built-in functionalities through NIW (NFV over IP over WDM) framework, to import or synthesize historic IP monitoring traces, apply filtering techniques and apply simple forecast regression models to predict future traffic growths.
2. Understand and gain key network planning capabilities related to the estimation of IP traffic matrices, from link count observations. Use Net2Plan built-in traffic matrix estimation functionalities in several use cases.

2 Duration

This lab work is designed for two sessions of two hours.

3 Evaluation

This lab work has been designed to guide the students in their learning of the subject. The annotations the students make in this document are for their use when studying the course, and do not have to be delivered to the teacher for evaluation.

4 Documentation

The resources needed for this lab work are:

- Lecture notes of the course.
- Net2Plan tool and their documentation (see <http://www.net2plan.com/>).
- Instructions in this wording.

5 Previous work before coming to the lab

- Read lecture notes related to traffic demand monitoring, forecasting and traffic matrix estimation.
- Refresh your reading in <http://www.net2plan.com/>, in particular, on monitor/forecast right click options in the demand and link tables.

6 Setting up the network scenario

Follow these instructions for setting up the network scenario:

1. Activate the NIW framework in Net2Plan, clicking the NIW button in the topology panel. NIW refers to NFV-over-IP-over-WDM, a framework that permits analyzing IP networks, where IP links can be transported as optical circuits on top of WDM networks, and where network functions can be virtualized and executed as VNFs instantiated in the data-centers in the nodes. In this lab work we will concentrate just on the IP part of the network.
2. Open Net2Plan and load the NSFNET network design in the file `niw_example7nodes_ipLinks.n2p`.
3. Set that the network behavior will be simulated as if the IP layer was governed by OSPF algorithm for all demands routed via hop-by-hop routing. For this, in the **What-if analysis** tab, activate the what-if analysis mode, and set as simulation algorithm the `StatelessSimulator_niw`, in the `BuiltInExamples.jar` code repository. Leave default options applicable. that only affect IP demands routed with source-routing (e.g. usable to simulate MPLS-TE connections, not a part of this lab).

Note that all the traffic is carried in the network, observing the **IP demands** tab.

4. We assume that link capacity is composed of groupings of 10 Gbps modules. In the **Links** table, set the link capacity to be the smallest multiple of 10 Gbps that makes link utilization to be lower or equal than 0.25. Use the right-click option **Set selected links capacity...** To match a given utilization.

All the traffic should be carried now. Fill in the next table with some relevant performance information.

Situation one year ago

Total offered traffic (Gbps)	
Total link carried traffic (Gbps)	
Total link capacities (10G modules / $\rho_{max} = 0.25$) (Gbps)	
Average end-to-end number of hops	

7 Synthesizing the demands' monitoring traces

In this section we will synthesize traffic traces using a random model. Note that in real networks, the traffic traces would come from a monitoring process and imported e.g. from the NMS (Network Management System) of the network.

Follow the next steps:

1. Select all the demands in the table, and use the right-click option **Add synthetic monitoring trace to selected elements**. Generate a traffic trace with initial date one year ago, one sample every hour (3600 seconds), 1 year of samples (8760), using current traffic as initial traffic, with exponential progression with CAGR 30% (0.3), a daily variation with a peak-to-valley factor of 10.0, the busy hour starting at 12:00 and lasting for 3 hours. Use a coefficient of variation for the added noise of 0.1.
2. Visually inspect the monitored traces in the **Monitoring** panel.

8 Creating traffic forecasts

In this section we will create traffic forecasts, and observe the effects in the forecasts of using percentile-like filters to keep the busy hour information.

8.1 Forecast without percentile-based pre-filtering

First, we make the traffic forecasts and capacity planning according to it, without removing the daily variations. Follow the next steps, while filling in Table 8.1:

1. Create a traffic predictor using an exponential fit of the traffic, using the appropriate right-click option.
2. Change the current date of the design to 6 months after today (this is done in the **View/edit network state** -> **Network** tab).
3. Make the offered traffic be that of the forecasted traffic using the right-click option **Set offered traffic as forecasted traffic for selected elements**. See that the forecasted traffic information is updated. Note that the relative mismatch column has now the zero value, since the forecasted and current traffic logically match.
4. Set the link capacities to be equal to the lowest multiple of 10 Gbps that provides a link utilization less or equal to 0.25.
5. Save the current design with the name `perDemandTraces_noPercentile.n2p`.

Non-percentile filtering

Exponential forecast variance explained (min % , max %)	
(Today forecast) Total offered traffic (Gbps)	
(6-month forecast) Total offered traffic (Gbps)	
(6-month forecast) Total link capacities (10G modules / $\rho_{max} = 0.25$) (Gbps)	
(6-month forecast) Average end-to-end number of hops	

Answer the following questions:

- Is the forecast accurate or even useful? Why?

- Would it be useful if the traffic did not have such steep day variations?

- Is the average number of hops changing? why?

8.2 Forecast with percentile-based pre-filtering

This section repeats the forecast process, but applying before a percentile-like filtering of the traffic. Fill in the next table with the required information, while progressing through the following steps:

1. Filter the monitoring information in all the demands, to keep one sample per day, with the 95% percentile traffic. How many monitoring samples exist now per demand? Visually inspect the traces. Is the new sampled data less noisy? why?

2. Create a traffic predictor using an exponential fit of the traffic, using the appropriate right-click option. Is the predictor more accurate now? (e.g. see the graph and the *variance explained* column)

3. Change the current date of the design to 6 months after today (this is done in the **View/edit network state** -> **Network** tab).
4. Make the offered traffic be that of the forecasted traffic using the right-click option **Set offered traffic as forecasted traffic for selected elements**. See that the forecasted traffic information is updated. Note that the relative mismatch column has now the zero value, since the forecasted and current traffic logically match.
5. In the **Links** table, set the link capacity to be the smallest multiple of 10 Gbps that makes link utilization to be lower or equal than 0.25. Which is the sum of the extra capacity needed in the links to cope with the traffic estimated for six months ahead?

6. Fill in the Table8.2.
7. Save the current design with the name `perDemandTraces_percentile.n2p`.

Percentile filtering

Exponential forecast variance explained (min % , max %)	
(Today forecast) Total offered traffic (Gbps)	
(6-month forecast) Total offered traffic (Gbps)	
(6-month forecast) Total link capacities (10G modules / $\rho_{max} = 0.25$) (Gbps)	
(6-month forecast) Average end-to-end number of hops	

Discuss why the forecast is more accurate now, to track the variations of the traffic along the different busy hours of consecutive days.

9 Estimating the traffic matrices

In this section we will address a form to forecast the future traffic matrix, making use of the link count traces, as if the traffic demand traces were not available.

First we will create the per-link monitoring information and forecast its evolution. Then, we will use link counts forecasted 6 months ahead, and estimate the traffic demands 6 months ahead from it.

9.1 Creating per-link monitoring information

Follow the next steps:

1. Load the file `perDemandTraces_noPercentile.n2p`, with the per-demand traffic traces.
2. In the **Links** table, create the link samples that would be monitored if the traffic changed according to the demand samples. For this, use the **Add link monitoring traces simulating effect of demand traces** right-click option.
3. Remove the per-demand monitoring samples, and the per-demand traffic predictors to reflect the situation that we have access to link counts, but not to per-demand information.
4. Filter the monitoring information in all the links, to keep one sample per day, with the 95% percentile traffic. How many monitoring samples exist now per link? Visually inspect the traces. Is the new sampled data less noisy? why?

5. Create a traffic predictor per link using an exponential fit of the traffic, using the appropriate right-click option. Which is the amount of variance explained? Inspect visually the estimation in the **Monitoring** panel.

6. Save the current design with the name `perLinkTraces_percentile.n2p`.

9.2 Estimations of per-demand traffic from link counts

Follow the next steps:

1. Load the file `perLinkTraces_percentile.n2p`. Set the design date to be **6 months from today**. In the **IP Demands** table, set the offered traffic to be equal to the forecasted traffic. Now, in the **IP Links** table, set the forecasted carried traffic to be the current carried traffic, with the option **Set forecasted traffic as constant = current traffic in selected elements**. Now, the forecasted traffic for the demands in the design is our ground truth traffic, and the forecasted traffic in the links is the one matching it. We will use this design to evaluate the performance of the traffic matrix estimation methods. Save the design as `tmEstim_groundTruth.n2p`. Fill in the column **GT** (ground-truth) in Table 9.2.

Note: Now the `tmEstim_groundTruth.n2p` design, has as IP demand forecasted traffic the ground-truth for six months ahead, and a traffic predictor for each *IP link*. This is the information we will use to benchmark the traffic matrix estimation procedures.

2. Estimate from the `tmEstim_groundTruth.n2p` design, the IP traffic demands using the forecasted link carried traffic for 6 months from today, using the methods below.

Note: All these methods, take the IP traffic forecasted per link, estimate the IP traffic matrix from it, and leave it as *offered traffic in the demands*. Then, we will have for each demand and IP link i) its estimated traffic in the *Carried traffic* columns, and ii) its ground-truth traffic in the *Forecast* columns.

- *GM*. Gravity model from current date link carried traffic forecast (`Forecast demands traffic using gravity model...` -> from current date link carried traffic forecast)
- *Reg-NR*. Regression method, with no input demand information. Use the method `Forecast demands traffic using regression from link forecast info`.
- *Reg-GM*. Regression method, using as regularization the demand values given by the gravity model (balance factor 0.5). For this, in the option `Input of demand monitoring`, choose `Use gravity model estimation from link forecast`.

Fill in the following table.

Percentile filtering (GT = Ground truth).

Performance (6-month forecast)	GT	GM	Reg-NR	Reg-GM	Reg-R+
Total offered traffic (Gbps)					
\sum demand absolute errors respect to GT (Gbps)					
\sum link absolute errors respect to GT (Gbps)					
Total link cap. (10G mods / $\rho_{max} = 0.25$) (Gbps)					
Average end-to-end number of hops					

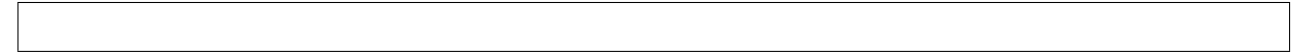
9.3 Improving the estimations of per-demand traffic

A form of improving the estimations is monitoring a small number of end-to-end flows, preferably the ones of a higher expected load. Then, this information can be introduced in the regression model together with the link counts, improving the estimation of the rest of the flows.

To test this method, follow the next steps:

- Load the `tmEstim_groundTruth.n2p` file.
- Remove the predictions of all but the 6 largest IP demands.
- Use the option `Forecast demands traffic using regression from link forecast info` -> `Current date demand forecast info` to improve the regression information with the demand forecasts available.

Fill in the column `Reg-R+` in the previous table. Are the results improved?



10 Work at home after the lab work

The student is encouraged to complete all the tasks that he/she could not finish during the lab session. Additionally, the following *Quizzes* are suggested.

Quiz 1. Observe the effect of changing the balance factor in the Reg-GM method, and how it affects the link and demand relative errors. Why?

Quiz 2. In the design, remove the link between Madrid and Valencia. This converts the network into a ring. Repeat the tests described above. Are the matrix estimation methods more precise?

Quiz 3. Repeat the previous tests but applying a CAGR factor in the traffic of 0.1 to half of the demands and 0.9 to the other half. Are the matrix estimation methods more precise?

Quiz 4. Repeat the previous tests but removing the daily busy hour effect in the demands when creating the monitored traces. Observe and discuss the effect of the per-day percentile filtering in this case.