Universitat de Girona
Escola Politècnica Superior

Treball final de màster

Estudi: Màster en Enginyeria Informàtica

Títol:
Simulador i visualitzador de robustesa de xarxes
(Network robustness simulator and visualizer)

Document:
Resum

Alumne:
Sergio Gómez Cosgaya

Tutor: Jose Luís Marzo Lázaro
Departament: Arquitectura i tecnologia de computadors
Àrea: Arquitectura i tecnologia de computadors

Convocatòria (mes/any)
Setembre 2017

Contents

Chapter 1

Introduction

1.1 Introduction, motivations and aims of the project

In a full communicated world where people around the world communicates each other through big networks it is essential to take care of these networks and make them available at much as possible. As a network must deal against attacks and failures, an analysis of them is mandatory, starting from checking its characteristics to executing failure simulations.

Related to network analysis, the research group **BCDS**[1] of University of Girona has been working for a while, so when I joined BCDS I started working on this area.

1.1.1 Motivations

In the research group, there was significant work done about network robustness analysis, that is analysing how a network is affected against several attacks. To analyse robustness, different experiments had to be executed with different kind of attacks and attack combinations which produced several experiments to execute.

This project is related to this previous work done and the main goal was to make these experiments easier. Previously, all executions had to be done manually and individually from a command line. Moreover, executions could not be executed in background, that means that who executed the experiments had to wait until executions finish and it was impossible to analyse results in a graphical way because they only had a static network visualizer which with a high number of nodes make the visual analysis impossible to achieve.

The main motivation was to start working in this topic and help the research group making an automatic tool or (from now referred as **simulator**) which could help them to execute their experiments easily. Also, another important motivation was to start working in a research group, learning how *research world* is.

1.1.2 Objectives

The main objective is to develop the automatic simulator which will be able to execute several experiments. The main functionalities it must have are:

- Simulator must be available worldwide through web access.
- Execute multiple experiments automatically from a single call.

[1] Broadband Network Control and Management. Official website: http://bcds.udg.edu

- Execute experiments in background, allowing the user to quit the simulator and executions do not stop.
- Full data results download availability.
- Nice results and networks visualizations.
- Simulator must have different workspaces to not merge different type executions and have users management.

Chapter 2

System requirements

In this section all system requirements that application must accomplish are detailed. Both functional and non-functional requirements have to be defined before any implementation. We differ about simulator and visualizer requirements.

2.1 Functional requirements

Functional requirements are those that represents every functionality the system has to be able to perform. They are based on an user input and expect an output or a result.

Simulator functional requirements are:

Register and login – Be able to register and log in using different social networks (using a no-password user system).

Administrator managing – An administrator must be able to manage all users, simulations and networks in the system.

Different workspaces – Have your own private workspace to work in his your experiments.

Shared workspaces – Be able to have shared workspaces with other users.

Compute simulations – Compute simulations over a network using some parameters and metrics.

Compute set of simulations – Compute a set of simulations at once without executing each of them individually.

Computing again – Compute a simulation or a set of them instantly from another simulation or set previously computed.

See computed simulations – Be able to see all simulations and sets previously executed. Simulations from a workspace can not be seen from another workspace or user.

Download results – Download simulations results in a file (off-line availability).

Preview results – Preview results dynamically on application (on-line availability).

Remove simulations – Remove a simulation or a set of them from the system.

Import networks from public repository – Be able to use public networks by importing them from a public repository available in the system. Public networks are available for any user.

Upload new networks – Be able to upload custom or personal networks to the system. These networks have to be only available for user who uploaded it.

Generate networks – Be able to generate several kind of networks and to import them to system. These generated networks have to be only available for user who generated it.

Visualizer functional requirements are:

View networks synthetically – Be able to preview any network formatted in GraphML file in a synthetic way, i.e. the network shape is defined dynamically using forces between nodes and edges.

View networks on a map – Be able to preview any network formatted in GraphML file on a map. Network nodes must have geographical attributes (e.g. coordinates).

Focus on mouse-over – Focus elements when pointing them with cursror, i.e. reducing other elements visibility or increasing pointed elements visibility.

Analyse attributes – Be able to see network, nodes and edges attributes. Also be able to colourise elements depending on their attributes values.

Filter elements – Show only desired elements hiding elements that not satisfy our needs (e.g. show only edges with bandwidth higher than 10Gbps).

Search elements – Find any elements matching with a keyword inside whole network.

Visual settings – Be able to modify all visual settings (e.g. change nodes color and size, nodes positioning, etc.).

Dynamic computing – Be able to execute dynamically different analysis over the network visualized at the moment (e.g. calculating the diameter in whole network).

2.2 Non-functional requirements

Non-functional requirements are those that system are supposed to do but are not directly a functionality of the system.

Simulator non-functional requirements are:

Stability – Application have to be stable, even when conditions are not optimal. System should not turn off for no reason.

Worldwide availability – System have to be accessible anywhere.

Execution performance – Execution performance has to be better as possible. That is adding availability to execute more than one simulation at the same time and to efficiently use all hardware resources (e.g. panellizing executions over all available threads).

User permissions – User can only see his experiments and networks or those ones shared with other users.

User enabling – User can not use his account just after registering because administrator has to enable the user before. That helps no one without permissions can have access to application.

Execution time-out – System can detect when an execution is no longer working (e.g. it has entered into a loop) and stop it from executing.

Unmerged code – Simulator code have not to be merged with execution code. When changing execution code should not have to affect to simulator.

Friendly interface – Simulator must have an easy and understandable interface that make user interaction and learning faster.

Chapter 3

Results

3.1 Developed application

Web application is available worldwide in research group servers[1] through login page with some social networks login. Then user is able to select a workspace and select parameters to compute a new experiment. When execution finished, we can see an experiment overview page with all simulations launched and their results. All previous selected parameters are shown and also a list with experiment simulations appears with ability to click on each of them to get details.

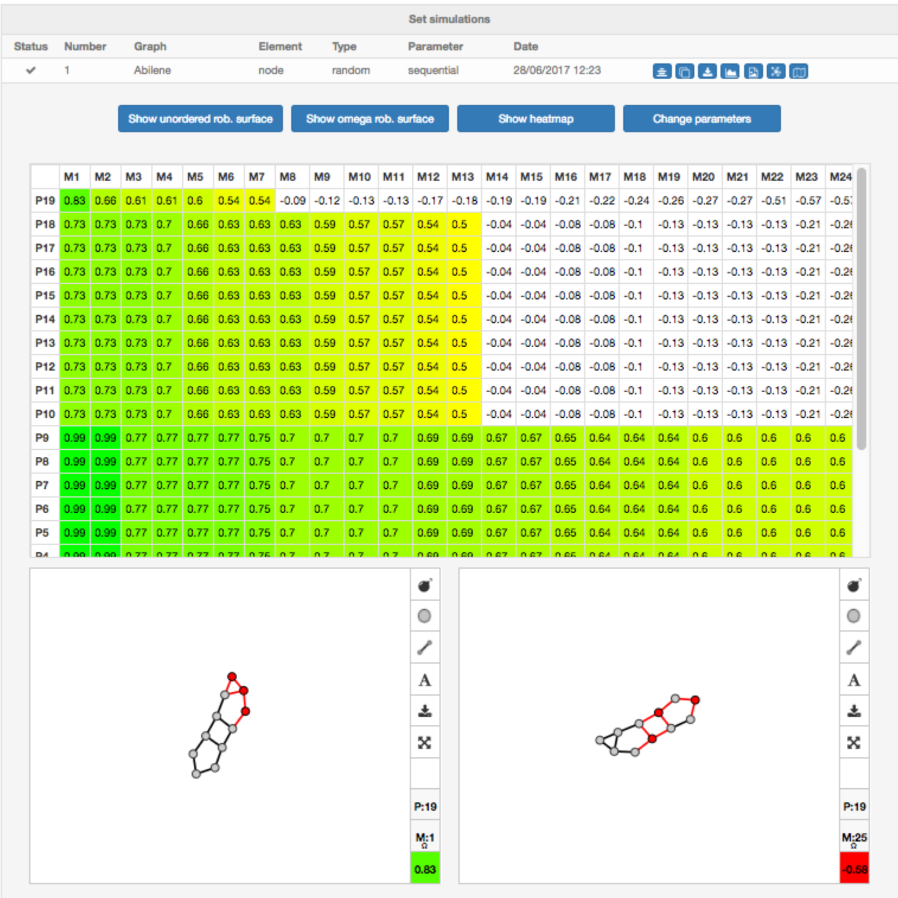

FIGURE 3.1: Single execution attacks comparison using visualizer.

[1] http://infraiiia01.udg.edu

Each simulation has its own buttons to see results in different ways as downloading XML files, seeing them in visualizer or seeing execution output. In figure 3.1 a screenshot of a comparison between two different generated attacks can be seen. Left side visualization shows a low damage attack (robustness remains at 0.83) and in right side visualization a high damage attack is shown (robustness value is less than 0), that is with same number of elements attacked (3) generated effect can differ considerably. That happens because in left side attack elements selected are not central but in right side one are. Application as detailed in the report has other functionalities than executing simulations. Users can visualize networks using the visualizer tool in both synthetic mode or over a map.

3.2 Collaborations

The main goal of this project was to achieve a useful tool but also it was important to find other research groups who want to use this tool in order to add their studies on it as collaborators. Other groups can use the application as a tool to visualize their complex networks or dynamical computations over them. Existing collaborations are detailed below:

3.2.1 Technical University of Munich (TUM) collaboration

The Department of Electrical and Computer Engineering of TUM has been working in routing algorithms using demand matrices through a network. They are able to find optimal paths for an existing network and demand matrix in order to maximize throughput and reduce link usage. TUM's work is based on SNDLib[2] networks and demand data. This algorithm has been added to our simulator tool as two new metrics (28-TRH and 29-LU). These two metrics can be categorized as functional metrics and define the amount of traffic lost after a failure in the network, that is removing an element and computing again the algorithm, checking if resulting throughput value remains equal than before (using different paths) or has a lower value (some traffic can not be handled).

3.2.2 Kansas State University (KSU) collaboration

Kansas State University research group has been working in epidemic models in order to understand how an infection or failure can spread over a network. KSU has elaborated an algorithm to generate attack sequences using these epidemics models and thanks to our collaboration we have added their code in our simulator to let it have the ability to generate dynamically epidemic attacks.

FIGURE 3.2: Epidemic attack spread example (from left to right). Next selected node has to be connected to an infected node.

[2]SNDLib is a library of test instances for Survivable fixed telecommunications Network Design. Official website: http://sndlib.zib.de/

3.2.3 Institut Català de Recerca de l'Aigua (ICRA) collaboration

ICRA is the most important research institution related to water in Catalonia. They work in topics about water distribution networks and sewerage system. In this case, we have not added any code in our simulator but they use the visualizer in order to visualize their big networks and to compute analysis over them. They shared with us the Girona's sewerage network and after some exhausting fixing, filtering and conversion done by us it is now accessible through the visualizer. Aside from visualizing network ICRA asked us for computing some analysis on this network as for example grouping pipelines depending on some criteria. We helped them adding this functionality to the simulator.

3.3 The 3rd Delft-Girona Workshop of robustness of complex networks

In the background of the 3rd Delft-Girona Workshop[3] related to complex networks robustness that took place in Delft University of Technology (The Netherlands) on 11th July, 2017 a depth demonstration of the whole application with a great acceptance from present public in workshop. A whole session was needed to explain all simulator functionalities (theoretical and practical demonstration).

Chapter 4

Conclusions

Having in mind the objectives of this project, these are the general conclusions:

- It has been **successfully developed the simulator** in a web application including its three tools as simulator, visualizer and generator. The application is able to compute experiments in a user friendly way and it is accessible worldwide.
- The previous existing R code has been changed to new web based system with **many functionalities added** that make the whole system more powerful. All functionalities are now accessible easily through a navigator which is helpful to researchers with no computing knowledge.
- Other **institutions and/or collaborators are interested in using this application** in order to use it for their experiments. They see the application as a very helpful one thanks to many features available on it that they don't have in their systems.
- Aside from using synthetic networks, during the project textbfsome real networks has been used in order to analyse them using the application, which indicates the application **can be used in many kind of real networks**.

[3]More info at: https://www.nas.ewi.tudelft.nl/rocn/index.html

Personally, I believe this project has resulted in a very useful application which help BCDS research group members and others to execute and understand their experiments easily. The most important personal conclusions are:

- This project was the start on introducing myself in research topics. I started this project with the aim to learn how research is and to improve on it. I am really satisfied in this point, being examples the collaboration with Technical University of Munich with a two weeks stay in it or the assistance as speaker in the 3rd Delft-Girona Workshop of robustness of complex networks. This gave me much experience in order to improve with my personal trajectory.

- An important aspect in a final project is to use whatever you learned during the master. In this project I have used a lot of knowledge learned during the master. It is important to understand that this project is a join of two different topics in computer engineering which are web developing and research investigation.

Finally, I am proud of making most of the whole application with no external help than my previous knowledge in web developing and forums available in internet. This includes setting up the server, developing the application and testing it.