Ramón Doallo
Manuel Arenaz
Patricia González (eds.)

XX Jornadas de Paralelismo

A CORUÑA · 16-18 DE SEPTIEMBRE DE 2009

Actas de las
XX Jornadas de Paralelismo

A Coruña, 16,17 y 18 de septiembre de 2009

Organiza:
Grupo de Arquitectura de Computadores
Universidade da Coruña

EDITORES:

Ramón Doallo Biempica
Manuel Arenaz Silva
Patricia González Gómez

ACTAS DE LAS XX JORNADAS DE PARALELISMO

ISBN: 84-9749-346-8

Depósito legal: C 2318-2009

Edita: Servizo de Publicacións, Universidade da Coruña
http://www.udc.es/publicaciones c/e: publica@udc.es

Imprime: Macrom. Tel. 981 135 577
http://www.macrom.es c/e: macrom@macrom.es

XX JORNADAS DE PARALELISMO

Comité de coordinación

Ramón Beivide Palacio (UC)
Jesús Carretero Pérez (UCIII)
José Duato Marín (UPV)
Inmaculada García Fernández (UAL)
Antonio Garrido del Solo (UCLM)
Emilio López Zapata (UMA)
Emilio Luque Fadón (UAB)
Pedro de Miguel Anasagasti (UPM)
Alberto Prieto Espinosa (UGR)
Francisco José Quiles Flor (UCLM)
Ana Ripoll Aracil (UAB)
Francisco Tirado Fernández (UCM)
Mateo Valero Cortés (UPC)
Victor Viñals Yúfera (UZ)

Comité Organizador (UDC)

Ramón Doallo Biempica
Patricia González Gómez
Manuel Arenaz Silva
Margarita Amor López
Diego Andrade Canosa
Iván Díaz Álvarez
Basilio B. Fraguela Rodríguez
Gonzalo Iglesias Aguiño
Guillermo López Taboada
María J. Martín Santamaría
Emilio Padrón González
Xoán C. Pardo Martínez
Gabriel Rodríguez Álvarez
José R. Sanjurjo Amado
Juan Touriño Domínguez

Entidades Patrocinadoras

Vicerreitorado de Investigación. Universidade da Coruña
Ministerio de Ciencia e Innovación. Gobierno de España
Consellería de Educación e Ordenación Universitaria. Xunta de Galicia
Concello da Coruña
Red Gallega de Computación de Altas Prestaciones
Red Mathematica Consulting & Computing de Galicia
HP España
Bull España

Entidades Colaboradoras

Facultade de Informática. Universidade da Coruña
Oficina de Cursos e Congresos. Universidade da Coruña
Turismo Coruña
Turgalicia

Entidades Organizadoras

Grupo de Arquitectura de Computadores. Universidade da Coruña
Sociedad de Arquitectura y Tecnología de Computadores (SARTECO)

Presentación

Quisiéramos aprovechar estas líneas para daros la bienvenida a las XX Jornadas de Paralelismo, que tienen lugar en A Coruña entre los días 16 y 18 de septiembre de 2009. La celebración de este evento supone una vez más, y ya van veinte, el encuentro de numerosos investigadores para intercambiar experiencias y presentar ponencias que reflejan su trabajo en el ámbito de la arquitectura de sistemas de computadores.

El número de asistentes a las Jornadas de Paralelismo ha ido creciendo con cada nueva edición. En este año se han recibido 119 artículos, y el número de participantes con el que contamos asciende a más de 170. Finalmente el programa ha quedado estructurado en más de 20 horas de trabajo, repartidas en 24 sesiones que cubren los temas siguientes: tecnologías grid y plataformas distribuidas; aplicaciones; redes y comunicaciones; arquitecturas, algoritmos y aplicaciones sobre aceleradores hardware; arquitecturas de procesador, multiprocesadores y chips multinúcleo; algoritmos y técnicas de programación paralelas; compilación para sistemas de altas prestaciones; docencia en Arquitectura y Tecnología de Computadores (ATC); y evaluación de prestaciones.

Contamos en esta edición con dos conferencias plenarias, impartidas por los profesores Mark Baker, de la Universidad de Reading (Reino Unido), sobre Cloud Computing y sus servicios, y Enrique Quintana-Ortí, de la Universitat Jaume I, sobre técnicas superescalares en la construcción de bibliotecas numéricas para procesadores multinúcleo y GPUs. Además tendremos la oportunidad de asistir a dos sesiones técnicas impartidas por HP España y Bull España, y a la asamblea de la Sociedad de Arquitectura y Tecnología de Computadores (SARTECO).

El programa de las Jornadas incluye también dos mesas redondas sobre asuntos que actualmente están provocando un intenso debate entre la comunidad universitaria. La primera de ellas trata sobre cómo trasladar las fichas del Grado y Master en Informática a los nuevos planes de estudio. La segunda mesa redonda abordará el tema de la transferencia de conocimiento universidad-empresa.

No queremos acabar esta presentación sin mostrar nuestro agradecimiento a todos los organismos y entidades, públicos y privados, que han colaborado en el desarrollo de estas Jornadas, en concreto, a la Universidade da Coruña, al Concello de A Coruña, a la Xunta de Galicia, al Ministerio de Ciencia e Innovación, a la Red Gallega de Computación de Altas Prestaciones, a la Red Mathematica Consulting & Computing de Galicia, a las compañías HP y Bull, y finalmente a todos los miembros del Grupo de Arquitectura de Computadores de la UDC que han brindado su apoyo para la organización de este congreso.

Finalmente, queremos agradecer vuestro interés y participación en estas XX Jornadas de Paralelismo, que esperamos cumplan con vuestras expectativas.

Comité Organizador de las
XX Jornadas de Paralelismo
A Coruña, 16-18 de Septiembre de 2009

Índice

Algoritmos y técnicas de programación paralelas

Aplicaciones

Arquitecturas de procesador, multiprocesadores y chips multinúcleo

Arquitecturas del subsistema de memoria y almacenamiento secundario

Arquitecturas, algoritmos y aplicaciones sobre aceleradores hardware

Compilación para sistemas de altas prestaciones

Computación de altas prestaciones sobre arquitecturas paralelas heterogéneas

Docencia en ATC

Evaluación de prestaciones

Redes y comunicaciones

Tecnologías grid y plataformas distribuidas

Automating the Modeling and Simulation Life Cycle of Mobile Ad hoc Networks

P. Pablo Garrido, Manuel P. Malumbres, Jesus Llor, Esteban Torres[1] , and Carlos T. Calafate[2]

Resumen— **The use of simulation has grown in the networking field, specially for Mobile Ad hoc Networks (MANETs). Simulation plays a very important role when evaluating routing protocols. However, the validity of published simulation results has been questioned by several authors. One of the main reasons exposed in the research literature is related to topology modeling, since the specific scenarios chosen have a great impact on simulation results. Most simulation studies usually pick a few randomly generated simulation scenarios leading to a lack of credibility of such works. Several works have been carried out recently in which the authors define a set of metrics and propose several recommendations for properly choosing the simulation scenarios. However, it is almost impossible to follow those recommendations without the support of specific tools, specially when network nodes are mobile. In this paper, a simulation framework is presented supporting both topology modeling and the simulation automation based on the OPNET Modeler simulator. The proposed set of tools is helpful at constructing rigorous MANET scenarios, as well as at performing other necessary tasks such as running simulations, extracting the selected data from the simulation results, and generating graphs and reports. These tools have been ported to the most common platforms, and can be executed either sequentially or in parallel on a Condor cluster.**

Palabras clave— **Modeling, Simulation Life Cycle, Automation, OPNET Modeler, Condor**

I. Introduction

Mobile Ad hoc Networks (MANETs) are composed by a group of wireless stations that communicate with each other to form a network. This kind of networks does not require any sort of infrastructure. Instead, a source node can communicate with a destination node directly (if both are within the communication range) or through a set of intermediate nodes that forward any incoming packet to the next neighbor in the established path (*multihop*).

Implementing a real MANET scenario is costly and both resource and time consuming. On the other hand, these networks are mathematically intractable [1]. So, most published research works about MANETs use simulation tools [2], but the reliability of such simulation studies has been questioned [3], [4], [5].

Modeling appropriate scenarios and realistic mobility patterns is an important task when simulating MANETs, since small differences in the scenario's topology have a large impact on routing algorithms, as stated in [6] and [7]. However, the scenarios are usually randomly generated in much of the research works without any kind of validation, leading to a lack of accuracy. So, a way to characterize and validate simulation scenarios prior to evaluating routing protocols is required. This problem has been widely addressed in the literature [2], [7], [8], [9], [10], [11]. For example, two metrics are used in [8] for constructing valid MANET scenarios in order to evaluate routing protocols: (1) average shortest path hop count, and (2) average network partitioning. Some metrics are defined in [10] in order to quantify the partitioning degree too.

However, putting into practice the suggestions given in such works is quite complicated without the help of adequate software tools. According to [2], some of the most commonly used simulator tools for modeling wireless networks are ns-2 [12], GloMoSim, and OPNET Modeler [13]. However, the OPNET Modeler simulator is not as extended as ns-2, and such kind of tools is not available. Thus, in this paper we present a simulation framework that is able to meet these research goals.

The remainder of the paper is organized as follows. Firstly, an overview of the developed simulation framework is presented in section II. In Section III, the modeling of MANET scenarios and the methodology for running a set of simulations are described. Section IV describes how to process the resulting data from the simulations in order to generate graphs and reports that make easier the analysis of such data. Finally, conclusions are drawn in Section V, along with references to future work.

II. Framework overview

This paper presents a set of tools developed specifically for the OPNET Modeler simulator in order to automate, not only the modeling, but also all the phases involved with the simulation cycle. The specific purpose of each one is briefly described on the following:

- *simul_topogen*: this is a topology generator that follows some of the recommendations proposed in the literature in order to build appropriate MANET scenarios.
- *simul_simul*: used for running one or several simulation sets, specifying, for example, different random number seeds or a list of values for some parameter.
- *simul_data*: used for extracting data from all the binary files obtained as result of running the simulation sets. This program invokes another

[1]Dpto. Física y Arquitectura Computadores, Univ. Miguel Hernández, e-mail: {pgarrido, mels, jllor, esteban.torres}@umh.es.
[2]Dpto. Informática de Sistemas, Univ. Politécnica Valencia, e-mail: calafate@disca.upv.es.

program called, *ov2txt*, which extracts data from the result file (.ov) for each individual simulation and saves it to a text file.
- *simul_graphs*: used for generating graphs from the extracted data.
- *simul_reports*: used for building a report for each programmed simulation set.

All these tools are command-line programs and were developed using the C programming language. This has allowed to migrate them easily to several platforms such as MS-Windows, UNIX-like systems (e.g. Linux), or MacOS X. However, running some of these programs will be only possible if the OPNET Modeler has been ported to that specific platform. Currently, it is available for MS-Windows and Linux (since v11.5). For example, the *simul_topogen* tool needs the External Model Access (EMA) package from OPNET in order to building MANET scenarios, and the *simul_simul* tool invokes the *op_runsim* program to run each simulation from the command line. The same applies to the *simul_data* program because the program *ov2txt* is invoked, requiring the EMA package too.

On the other hand, the remainder tools do not need the OPNET Modeler simulator, and so they can be executed on a different platform if necessary. In particular, *simul_graphs* can be executed on any platform in which either the R package [14] or gnuplot [15] are available. About the *simul_reports* tool, this only needs a LaTeX[16] distribution, which is available in many platforms.

Finally, all the developed tools are able to work both sequentially and in parallel when a cluster system is available (e.g. Condor [17]). This is achieved easily by specifying a simple option from command-line, drastically reducing the time spent in each step of the whole process.

III. Modeling and Simulation

A. Modeling MANET scenarios

Network simulators typically receive a scenario as input, which, in the particular case of MANETs, describes the mobility pattern of a group of nodes. Both the OPNET Modeler simulator and the proposed *simul_topogen* utility allow the user to randomly set the initial node placement within a specific area, as well as setting a mobility profile.

With regard to the mobility pattern of nodes, several mobility models have been proposed [18], but only a few are available in OPNET Modeler. With OPNET Modeler it is possible to use either a centralized or a decentralized approach when modeling the mobility pattern. In the centralized approach, a common mobility profile is used for all the nodes, and a 'Mobility Config' object is needed. This object holds the random mobility parameters for the Random Waypoint mobility model (RWM), namely, the area of the movement, the speed, and the pause time. This is the most common mobility model for MANETs [8], [19]. The random path followed by each mobile node is recorded in an ASCII text file (with a .trj extension) by running a simulation. If more flexibility is required, the decentralized approach is more adequate, allowing to define a distinct profile for each node.

When executing one simulation, one random number seed must be specified. Thus, to ensure the statistical validity of the obtained results, it is necessary to repeat the experiment with distinct random seeds. However, changing the seed affects the trajectory followed by the mobile nodes. In order to use the same trajectory independently of the random number seed used, the generated trajectory file for each node using the centralized approach should be assigned to the corresponding mobile node using the 'trajectory' attribute. OPNET Modeler offers an option to do this assignment.

Although, all the aforementioned options within OPNET Modeler are useful, when it is necessary to generate a great number of MANET scenarios, doing it manually can be a tedious task and prone to error. Our *simul_topogen* tool does all those steps automatically for a specified list of seed values, generating a distinct scenario and trajectories for each one. Another drawback of manually doing these tasks within the OPNET Modeler environment is that there is no way of knowing if the scenarios generated are valid for conducting a rigorous research, specially in the presence of mobility. The position of nodes is a critical factor because one or more of them could be isolated. This is specially true in the case of static scenarios where the position of nodes provokes network partitioning. For example, if some nodes are isolated, they will not be able to communicate with each other during all the simulation time nor receive broadcast messages. In an dynamic scenario, this problem can also occur at any time due to the mobility of the nodes, but it is usually temporary.

Due to these issues, some researchers have proposed some recommendations when choosing the scenarios for simulation. For example, as stated in [8], in order to rigorously test a MANET routing protocol, the scenario should meet two standards: (1) the average shortest-path hop count needs to be large, and (2) only a small amount of network partitioning should exist. Therefore, the proposed *simul_topogen* tool takes into account such recommendations and checks the validity of any generated scenario.

The *simul_topogen* tool computes the average shortest-path hop count and several metrics defined in [8] and [10] in order to determine the network partition degree for each scenario generated. In short, the four metrics considered in *simul_topogen* are defined as follows:

- Average shortest-path hop count (AHC): the smallest number of hops along any path between each pair of nodes, taking into account the transmission range. The minimum value is 1, indicating that every pair of nodes are direct neighbours. Although the average number of hops is more commonly used when evaluating the performance of a protocol, this metric is more convenient to check for the validity of the generated scenarios because it is not protocol dependent.

- Average Network Partitioning (ANP): the proportion of node pairs with no communication path available. The optimum value for this metric is 0%, indicating that there is no partitioning degree and so all the nodes are able to communicate with each other.
- Average Number of Groups (ANG): the number of partitions that exist in the network. If this value is 1, all nodes are in the same group, that is, there is no isolated node.
- Average Partition Size (APS): the average number of nodes in each partition. The greater the size of a partition, the higher connectivity and lower partitioning degree of the network. If this value is exactly the number of nodes, all of them are in the same group. On the other hand, if this value is 1, each node is isolated from the others.

All these metrics are computed not only in the initial state, but also during the simulation time, being defined as the average value. Finally, the best scenarios are chosen in order to meet these two standards. The essential characteristics of the *simul_topogen* tool are shown by means of the following example:

```
simul_topogen -p myproject -m mynetwork
   -seeds    1 50 1
   -nodes    MOB node*
   -area     0 0 1900 400 0 0
   -mobility 5 0 0 0 1
   -checkpartition 250
   -duration 300
   -select   10 ANP ASC n
   -export   rh ..\trajectories
```

In this case, the template scenario to be tested is read from the 'mynetwork' model within the OPNET Modeler project 'myproject'. The values following the -seed argument are the seeds to be used, that is, the number of scenarios to be tested: from 1 to 50 by increments of 1 (50 seeds). Then, the network nodes to be randomly placed are specified; they can be either mobile nodes (MOB) or fixed ones (FIX), and those which name starts with 'node'. The specified nodes will be placed in a rectangular flat area sized 1900x400 meters. A 3D volume can be set too specifying the altitude range in the last two values of the -area parameter. Then, the RWM mobility model parameters are specified with the -mobility option in this order: speed, pause time, start time, stop time, and frequency. In order to check the network partitioning degree, the communication range is specified in the -checkpartition argument (250 m.). The simulation time (-duration) is set to 300 s. Then, with the -select option, the scenarios to be choosen are specified; in this case, they will be sorted by one of the evaluated metrics (ANP) in ascending order, and only the top 10 scenarios out of 50 generated will be selected, discarding those scenarios which present more than one partition at the initial state.

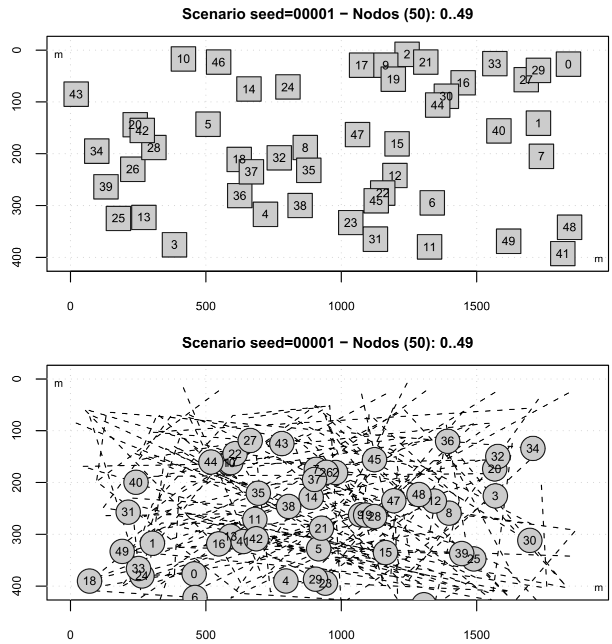

Fig. 1. Snapshot during a simulation (top) and trajectories followed by nodes (bottom)

Finally, the trajectories followed by each node can be exported to a single ASCII file readable either by the R package or by MS-Excel. The destination path for this file can be specified too. In addition, the *simul_topogen* utility is able to generate a set of graphical files showing the position of the nodes at any time and for each of the selected scenarios, that is, all the snapshots along the simulation time. Furthermore, the trajectories followed by all the nodes or a group of them can be exported to graphical files too, by post-processing the trajectory file generated previously. So, the user can choose any of the graphs generated for including them into a report or research paper outside of OPNET Modeler simulator (see Figure 1). However, OPNET Modeler includes an animation viewer that can be used to visualize the movement of the nodes during the simulation. All these facilities could be very useful in order to implement and verify a new mobility model, in a similar way than the iNSpect [20] utility does.

Depending on the simulation study to be conducted, the researcher could need to choose the best or worst scenarios. This can be done by simply specifying any of the computed metrics and the order of sorting (ASC or DESC) in the -select option.

B. *Running simulations*

Within the OPNET Modeler GUI, when the user runs a simulation, the *op_runsim* program is launched in the background. This program can also be executed from the command-line too by the user, specifying the appropriate arguments. However, this should be done for every simulation within the whole simulation set, that is, varying the random number seed, the scenario, traffic load, etc.

In order to automate this manual process, and to avoid human errors, the program *simul_simul* has been developed. The purpose of this tool is to run a set of simulations outside the OPNET Modeler environment in an unattended manner, either sequentially or concurrently. The program receives an ASCII text file as input, including the different parameters for the simulations to run, and these are launched by invoking the *op_runsim* program with the appropriate arguments each time. The usage of this program is the following:

```
simul_simul -f simuls.txt
   [-cluster 20] [-distributed]
```

The input file `'simuls.txt'`, which is specified through the `-f` option, is a tab-delimited file, and so it can be edited within the MS-Excel application for convenience. Each line corresponds to an independent simulation set, and each column to the arguments to the distinct simulations within the simulation set. Columns include mainly a list of random number seeds, and several lists of comma-separated values, which will be the varying arguments between simulations. Usually only one of these arguments is the varying value, being the value shown at the abscissas axis in the graphs (e.g. traffic load).

Optionally, if a Condor cluster system [17] is available, the simulation set could be run concurrently specifying the option `-cluster` along with the maximum number of simultaneous simulations, which has to be less or equal than the number of the cluster processors available. However, since each simulation needs the appropriate license, this value is limited by the number of OPNET licenses available too. The user only needs to specify the `-cluster` option, and the program automatically generates the necessary submit job files to launch all the simulations. That is, the user does not need any knowledge about the Condor system.

This should not be confused with the *Simulation Partitioning* concept [21], in which a single simulation is distributed onto a number of different machines in order to achieve a greater simulation scale than on a single machine. In fact, the OPNET Modeler has a preference that can be enabled or disabled to specify whether each simulation should be run sequentially or in parallel on several processors. By default, this preference is disabled; to enable the parallel preference from the *simul_simul* tool, the option `-distributed` should be specified in the command line.

IV. Result processing

A. Extracting data

All the statistics resulting from the OPNET Modeler simulations are stored in a single output vector file (file type suffix .ov). This is a binary file that must be converted to a text file in order to be imported and processed by external software like the R package, gnuplot, or MS-Excel, for example, in order to graphically present the data (see Section IV-B).

OPNET Modeler allows the user to export the collected statistics in three different ways:

1. Executing the 'Export Data to Spreadsheet' operation after showing a specific statistic within the Results Browser. By using a graph template several statistics can be exported at once; however this process takes a long time when dealing with a great number of simulations.
2. Using the `op_cvov.exe` and `op_cvos.exe` utilities available within OPNET Modeler since v14.0. Unfortunately, these are not available for previous versions.
3. By means of a self-designed C program which uses the External Model Access (EMA) package. This is the most convenient option because the process has been customized in order to be totally unattended, and it is available for any OPNET Modeler version.

Two different tools were developed for this purpose: (1) *ov2txt*, which uses the EMA package and exports several specified statistics from the output vector file (.ov) to a single ASCII text file, and (2) *simul_data*, which invokes to *ov2txt* for each simulation in a simulation set. Although all this functionality could have been implemented in a single program, doing it in this way allows to parallelize the process using a cluster, as it will be explained later.

The use of the *simul_data* program is very simple, since it receives exactly the same input file than *simul_simul* (see Section III-B). For example:

```
simul_data -f simuls.txt -x @stats.txt
   [-cluster 96]
```

This program interprets the simulations set defined in the input file '`simuls.txt`' and then iterates building the appropriate arguments for *ov2txt* invocation for each simulation. As it can be seen from the example, in addition to the input file it is necessary to specify the statistics to be extracted with the `-x` option. Although it is possible to give a list of numbers, the most convenient way is to specify a file name which contains the names of the selected statistics, as appear within the OPNET Modeler environment. For example, the file '`stats.txt`' contains the following:

```
time (sec.)
Wireless LAN.Throughput (bits/sec)
Wireless LAN.Delay (sec)
AODV.Number of Hops per Route
```

Finally, this program can do its task in parallel if the `-cluster` option is specified. In this case, each invocation to *ov2txt* will be a process, and the maximum number of simultaneous processes will be limited by the number of processors available and by the value specified in the command-line. Again, it is not necessary for the user to know anything about how to launch jobs in Condor, as this program generates all the necessary job files and submits them.

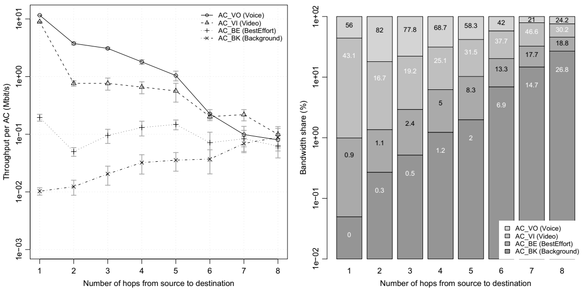

Fig. 2. Examples of graphs generated by *simul_graphs*

B. Generating graphs

The *simul_graphs* tool focuses on generating graphs from the data extracted previously. All the exported data for any individual simulation within a simulation set are taken into account. The usage of *simul_graphs* is very similar to the previous ones:

```
simul_graphs -f simuls.txt graphs.txt
   [-ci 90] [-gnuplot] [-cluster 96]
```

Again, `'simuls.txt'` is the simulation file explained before, and `'graphs.txt'` is the graphs file, which is another text file describing each graph to be generated. If the `-ci` option is specified, confidence intervals will be computed and shown in the plots using the confidence level specified (90, 95, 99, ...).

The format of the graphs file is very similar to the simulations one because it is an ASCII text tab-delimited file. So, both can be edited as a spreadsheet for convenience. The most relevant data included in the graphs file are: title and subtitle, the x and y axis titles, the column number for the abscissa values within the data files, the column numbers for any of the data series to be plotted and the corresponding serial names to be included in the legend. Additionally, the limits of the graph, the position of the legend, and the kind of graph to be generated. Currently, only two graph types are supported: lines and stacked bars (see Figure 2).

The graphs are generated by means of some predefined and parameterized R scripts [14], thus the user does not need to know the R language to obtain the graphs. However, the user can modify the R scripts in order to custom the output if needed. Each graph is generated in two graphic formats, namely, Encapsulated PostScript (EPS) [22] and Portable Network Graphics (PNG) [23]. The EPS graph files are vectorial format and it is the recommended format to be included when writing a research paper, whereas the PNG graphs are recommended to be included in MS-Power Point presentations.

On the other hand, if `-gnuplot` is specified, gnuplot [15] script files will be generated and executed. In this case, the same graphs will be generated again, but with a distinct appearance. The formats generated are: EPS, PNG, or Scalable Vector Graphics (SVG) [24]. We are planning to include more distinct outputs (e.g., Matlab or xgraph). In this way the user can choose the desired graphics package, and modify the automatically generated scripts without editing and recompiling the software.

As the previous programs, *simul_graphs* can be executed in parallel in a Condor system by specifying the `-cluster` option, drastically reducing the time spent in this process. In this case, each invocation to a R or gnuplot script file will become a process in the Condor system.

C. Generating reports

Finally, the last program presented allows to put all the graphs and the summarized data used for each graph together in a single document. That way, the user can analyze the results in both a visual and in a numerical way more easily. The program in charge of doing this task is *simul_reports*, whose usage is the following:

```
simul_reports -f simuls.txt graphs.txt
  [-outputprofile {pdf|dvi|dvi-ps|dvi-ps-pdf}]
```

where `'simuls.txt'` is the same simulation file used in all the previous steps, and `'graphs.txt'` the same graphs file used for *simul_graphs*.

The result of this program is a single document per simulation set, generated using the LaTeX document preparation system [16]. Firstly, the *simul_reports* builds a source file (with a .tex extension) on the fly. Then, depending on the output profile specified with the `-outputprofile` option, the source file is compiled into one of the following formats: Portable Document Format (PDF, by default), Device Independent format (DVI), Postscript (PS) or PDF through Postscript. There exist viewers for all of these file formats for the most common platforms. The graphical format included in the final document (e.g. PNG or EPS) depends on the output profile chosen, but all of them must have been generated previously by *simul_graphs*. Finally, if the user knows the LaTeX language, the source file generated can be modified or included in his/her own documents.

V. Conclusions and future works

Although simulation is the most common technique used for testing MANET routing protocols, it is necessary to validate the scenarios generated. Some recommendations and metrics have been proposed in the literature in order to characterize simulation scenarios. However, putting such suggestions into practice is very difficult without the support of adequate software tools. The framework presented in this paper is very effective and useful when simulating MANET experiments using the OPNET Modeler simulator.

The first of the simulation tools is a topology generator which considers some of the metrics suggested. The best scenarios (or the worst ones) of a set of randomly generated scenarios can be chosen, depending on one of the metrics evaluated. More metrics could be easily added if needed. In addition, it also plots the selected scenarios, including both the initial state and the trajectories followed by each mobile node.

The remainder tools aim at simplifying the simulation process and the subsequent analysis of results. They provide utilities for automating tasks such as running simulations, extracting relevant data, generating graphs, and building reports including both the data and graphs previously generated. All these tasks can be done outside the graphical interface of OPNET Modeler, in batch mode on the most common platforms.

About the usability of these tools, although all of them are used from command-line and there is no graphical interface, they are very easy to use, as their syntax is very similar. Moreover, the tools support the use of clusters of workstations for parallel processing, improving dramatically their running times. Finally, the user does not need to know anything about the R package, gnuplot, LaTeX, nor the Condor system. Thus, this framework has a weak learning curve.

As future work, it could be interesting to be able to export the topology and trajectory files generated (.trj files) by *simul_topogen* to other simulators like ns-2. This would allow to compare the simulation results against other simulators, further validating the obtained results.

The set of tools can be downloaded from the Gatcom web site (`http://atc.umh.es/gatcom/`), although they are still under active development.

Acknowledgments

This work was partially supported by the Ministerio de Educación y Ciencia, Spain, under Grant DPI2007-66796-C03-03. The authors also would like to acknowledge OPNET Technologies, Inc. for providing an educational OPNET Modeler license.

Referencias

[1] L. Felipe Perrone, Yougu Yuan, and David M. Nicol, “Modeling and simulation best practices for wireless ad hoc networks,” in *Proceedings of the 35th Winter Simulation Conference (WSC)*, 2003, pp. 685–693.

[2] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso, “Manet simulation studies: the incredibles,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 4, pp. 50–61, 2005.

[3] David Cavin, Yoav Sasson, and Andre Schiper, “On the accuracy of MANET simulators,” in *2nd ACM int. workshop on Principles of Mobile Computing (POMC)*, 2002, pp. 38–43.

[4] K. Pawlikowski, H. D. J. Jeong, and J. S. R. Lee, “On credibility of simulation studies of telecommunication networks,” *IEEE Communications Magazine*, vol. 40, no. 1, pp. 132–139, 2002.

[5] Todd Andel and Alec Yasinsac, “On the credibility of manet simulations,” *Computer*, vol. 39, no. 7, pp. 48–54, July 2006.

[6] G. Amoussou, B.L. Agba, Z. Dziong, M. Kadoch, and F. Gagnon, “Performances analysis of mobile ad hoc routing protocols under realistic mobility and power models,” in *OPNETWORK 2006*, Washington D.C, USA, 2006.

[7] M.B. Doar, “A better model for generating test networks,” *Global Telecommunications Conference, 1996. GLOBECOM ’96. ’Communications: The Key to Global Prosperity*, pp. 86–93, Nov 1996.

[8] Stuart Kurkowski, Tracy Camp, and William Navidi, “Two standards for rigorous manet routing protocol evaluation,” in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, Oct. 2006, pp. 256–266.

[9] S. Kurkowski, W. Navidi, and T. Camp, “Discovering variables that affect manet protocol performance,” in *Global Telecommunications Conference, 2007. GLOBECOM ’07. IEEE*, 26-30 Nov. 2007, pp. 1237–1242.

[10] J. Hahner, D. Dudkowski, P.J. Marron, and K. Rothermel, “Quantifying network partitioning in mobile ad hoc networks,” in *Mobile Data Management, 2007 International Conference on*, 1-1 May 2007, pp. 174–181.

[11] H. Haddadi, M. Rio, G. Iannaccone, A. Moore, and R. Mortier, “Network topologies: inference, modeling, and generation,” *Communications Surveys & Tutorials, IEEE*, vol. 10, no. 2, pp. 48–69, Quarter 2008.

[12] “The Network Simulator – ns-2,” `http://www.isi.edu/nsnam/ns/`.

[13] Opnet Technologies Inc., “OPNET Modeler,” `http://www.opnet.com/`.

[14] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2006, ISBN 3-900051-07-0.

[15] Jeff Racine, “gnuplot 4.0: a portable interactive plotting utility,” *Journal of Applied Econometrics*, vol. 21, no. 1, pp. 133–141, 2006.

[16] Leslie Lamport, *LaTeX User’s Guide and Document Reference Manual*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[17] Condor, “Condor project,” `http://www.cs.wisc.edu/condor/`.

[18] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[19] J. Yoon, M. Liu, and B. Noble, “Random waypoint considered harmful,” in *IEEE INFOCOM Conference*, 2003.

[20] S. Kurkowski, T. Camp, N. Mushell, and M. Colagrosso, “A visualization and analysis tool for ns-2 wireless simulations: inspect,” in *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005. 13th IEEE International Symposium on*, 27-29 Sept. 2005, pp. 503–506.

[21] Donghua Xu and Mostafa Ammar, “Benchmap: Benchmark-based, hardware and model-aware partitioning for parallel and distributed network simulation,” in *MASCOTS ’04: Proceedings of the The IEEE Computer Society’s 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, Washington, DC, USA, 2004, pp. 455–463, IEEE Computer Society.

[22] Peter Vollenweider, *Encapsulated PostScript: application guide for the Macintosh and PCs*, Prentice Hall International (UK) Ltd., Hertfordshire, UK, UK, 1990.

[23] T. Boutell, “Png (portable network graphics) specification version 1.0,” Tech. Rep., United States, 1997.

[24] Andrew Watt, *SVG Unleashed*, Pearson Education, 2002.

[25] K.I. Calvert, M.B. Doar, and E.W. Zegura, “Modeling internet topology,” *Communications Magazine, IEEE*, vol. 35, no. 6, pp. 160–163, Jun 1997.

XX Jornadas de Paralelismo

A CORUÑA 16-18 SEPTIEMBRE 2009

CERTIFICADO DE PONENCIA

D. / Dña. *GARRIDO ABENZA, PEDRO PABLO*

ha presentado el artículo titulado

AUTOMATING THE MODELING AND SIMULATION LIFE CYCLE OF MOBILE AD HOC NETWORKS

en las XX Jornadas de Paralelismo, celebradas en A Coruña del 16 al 18 de septiembre de 2009.

Universidade da Coruña – Departamento de Electrónica e Sistemas

Ramón Doallo Biempica
COMITÉ ORGANIZADOR

UNIVERSIDADE DA CORUÑA

Grupo de Arquitectura de Computadores
Facultade de Informática - Universidade da Coruña

Campus de Elviña s/n - 15071 A Coruña
web: http://gac.des.udc.es

XX Jornadas de Paralelismo

A CORUÑA 16-18 SEPTIEMBRE 2009

CERTIFICADO DE ASISTENCIA

D. / Dña. *GARRIDO ABENZA, PEDRO PABLO* ha participado

en las XX Jornadas de Paralelismo, celebradas en A Coruña del 16 al 18 de septiembre de 2009.

Universidade da Coruña
DEPARTAMENTO DE ELECTRONICA E SISTEMAS

Ramón Doallo Biempica
COMITÉ ORGANIZADOR

UNIVERSIDADE DA CORUÑA

Grupo de Arquitectura de Computadores
Facultade de Informática - Universidade da Coruña

Campus de Elviña s/n - 15071 A Coruña
web: http://gac.des.udc.es